Acoustic Models for Posterior Features in Speech Recognition

THÈSE N^{O} 4164 (2008)

PRÉSENTÉE LE 19 SEPTEMBRE 2008

À LA FACULTE SCIENCES ET TECHNIQUES DE L'INGÉNIEUR

Laboratoire de l'IDIAP

SECTION DE GÉNIE ÉLECTRIQUE ET ÉLECTRONIQUE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Guillermo ARADILLA

European Master of language and speech, Universitat politecnica de Catalunya, Espagne
et de nationalité espagnole

acceptée sur proposition du jury:

Prof. J. R. Mosig, président du jury
Prof. H. Bourlard , directeur de thèse
Prof. C. Nadeu, rapporteur
Prof. S. Renals, rapporteur
Prof. M. Unser, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Lausanne, EPFL
2008

Abstract

In this thesis, we investigate the use of posterior probabilities of sub-word units directly as input features for automatic speech recognition (ASR). These posteriors, estimated from data-driven methods, display some favourable properties such as increased speaker invariance, but unlike conventional speech features also hold some peculiarities, such that their components are non-negative and sum up to one. State-of-the-art acoustic models for ASR rely on general-purpose similarity measures like Euclidean-based distances or likelihoods computed from Gaussian mixture models (GMMs), hence, they do not explicitly take into account the particular properties of posterior-based speech features. We explore here the use of the Kullback-Leibler (KL) divergence as similarity measure in both non-parametric methods using templates and parametric models that rely on an architecture based on hidden Markov models (HMMs).

Traditionally, template matching (TM)-based ASR uses cepstral features and requires a large number of templates to capture the natural variability of spoken language. Thus, TM-based approaches are generally oriented to speaker-dependent and small vocabulary recognition tasks. In our work, we use posterior features to represent the templates and test utterances. Given the discriminative nature of posterior features, we show that a limited number of templates can accurately characterize a word. Experiments on different databases show that using KL divergence as local similarity measure yields significantly better performance than traditional TM-based approaches. The entropy of posterior features can also be used to further improve the results.

In the context of HMMs, we propose a novel acoustic model where each state is parameterized by a reference multinomial distribution and the state score is based on the KL divergence between the reference distribution and the posterior features. Besides the fact that the KL divergence is a natural dissimilarity measure between posterior distributions, we further motivate the use of the KL divergence by showing that the proposed model can be interpreted in terms of maximum likelihood and information theoretic clustering. Furthermore, the KL-based acoustic model can be seen as a general case of other known acoustic models for posterior features such as hybrid HMM/MLP and discrete HMM. The presented approach has been extended to large vocabulary recognition tasks. When compared to state-of-the-art HMM/GMM, the KL-based acoustic model yields comparable results while using significantly fewer parameters.

Keywords: *Automatic Speech Recognition, posterior-based speech features, template matching, hidden Markov models, Kullback-Leibler divergence.*

Résumé

Dans cette thèse, nous étudions l'utilisation des probabilités a posteriori des unités sous-mots directement comme caractéristiques d'entrée automatique pour la reconnaissance de la parole. Cettes probabilités, estimées à partir des données d'entrainement, affichent certaines propriétés favorables pour la reconnaissance de la parole, telles que l'invariance de l'orateur. De l'autre côté, ce genre de caractéristiques ont des certaines particularités, telles que leurs composants ne sont pas négatifs et de somme un. Les modéles état de l'art pour la reconnaissance de la parole sont basées sur des mésures générales comme, par example, la distance Éuclideenne ou des vraisemblances obtenus á partir des GMMs. Donc, ils n'exploitent pas les propiétés des caractéristiques basées sur des probabilités. Dans cette thése, on utilise la divergence Kullback-Leibler (KL) comme mésure de similitude dans des méthodes paramétriques (HMMs) et non paramétriques (TM).

Traditionnellement, les méthodes non-paramétriques utilisent caractéristiques basées sur le spectrum du signal obtenu de la parole. Donc, ils utilisent une grande quantité des examples pour décrire la variance de chaque mot. Ainsi, TM est généralement orientés au tâches dependantes de l'orateur avec un vocabulaire petit. Dans notre travail, nous utilisons les caractéristiques basées sur des probabilités. Compte tenu de la nature discriminatoire de ces caractéristiques, nous montrons qu'un nombre limié d'examples pour caractériser chaque mot. Les expériences sur des différentes bases de données montrent que l'utilisation de KL divergence comme mesure locale obtient de meilleures performances que les traditionnelles TM de manière significative. L'entropie peut également łtre utilisé pour améliorer encore les résultats.

Dans le cadre de HMMs, nous proposons un nouveau modèle acoustique o chaque état est paramétré par une distribution multinomiale référence et l'état score est basé sur la divergence KL entre la référence et les caractéristiques de la parole. Outre le fait que la divergence KL est une mesure naturelle entre les distributions a posteriori, nous motivons également l'utilisation de la divergence KL en montrant que le modèle proposé peut łtre interprété en termes de maximum de vraisemblance et du domaine de l'information théorique. En outre, le KL-acoustique basée modèle peut łtre considéré comme un cas général d'autres modèles acoustiques connus pour probabilités tels que hybrid HMM/MLP et discret HMM. Le présent approche a été étendue tches de reconnaissance avec un grand vocabulaire. Par rapport l'état de l'art HMM/GMM, le modèle basé sur la

distance KL obtient des résultats comparables tout en utilisant beaucoup moins de paramètres.

Mots-clés : *Réconnaissance automatique de la parole, caractéristiques du signal basées sur des probabilités, modèle correspondant, chaines cachées de Markov, divergence Kullback-Leibler.*

Acknowledgements

- ... Qu'est-ce que signifie "apprivoiser" ?

- C'est une chose trop oubliée, dit le renard. Ça signifie "créer des liens..."

- Créer des liens ?

- (...) si tu m'apprivoises, nous aurons besoin l'un de l'autre. Tu seras pour moi unique au monde. Je serai pour toi unique au monde...

Chapitre XXI - Le petit prince,

Antoine de Saint-Exupery

Doing a thesis is a long way. During this period, I have had ups and downs and I have had the chance of meeting many interesting people. Like in my case, their thesis have also helped them to better know themselves. We have walked together a part of our lives and, certainly, I would not be the same without having meet them. Thus, I feel I must sincerely acknowledge these life partners.

I have special sympathy for those anonymous people who do not speak too loud. They are not noticed by their words but by their silences, which I will probably remember the most.

Thank you, Deepu.

Thank you, Fabien.

Thank you, Faouzi.

Thank you, Hemant.

Thank you, Tamara.

Also, a little word for my supervisor, who gave all the freedom I needed for doing what I liked.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Objective of the Thesis

Given the "optimal" classification properties of posterior probabilities, the estimation of this type of measures has recently gained more and more attention in the field of automatic speech recognition (ASR). In particular, posterior probability estimates of sub-word units can be used for characterizing the speech signal in ASR systems. This speech representation can then be seen as a transformation holding some convenient properties for ASR, such as of being discriminative, minimizing the classification error and being invariant to the speaker and the environment. However, these same properties also make posterior-based speech features difficult to exploit in state-of-the-art ASR systems, hence, they are often used after specific "ad-hoc" transformations and need to be appended to standard features to yield improved performance.

In this thesis, we are developing principled approaches aiming at dealing directly with posterior probabilities as inputs for ASR systems. No specific transformation is thus needed and the properties of posterior-based features can be fully exploited. This is applied to improve the two main ASR approaches, i.e., non-parametric methods using templates and parametric models based on the hidden Markov model (HMM) architecture.

1.2 Automatic Speech Recognition

The goal of ASR is to recognize the message expressed by a spoken utterance independently of the speaker and the environment. Although first attempts in ASR research were based on knowledge-driven approaches (Klatt, 1977; O'Brien, 1993), current directions are based on statistical approaches that attempt to model a sequence of feature vectors extracted from the speech signal. These ASR approaches can be mainly classified into non-parametric and parametric methods. The former technique is known as template matching (TM) (Sakoe and Chiba, 1978; Bridle *et al.*, 1983) and describes each linguistic unit, e.g. words, using a set of feature sequences (templates) from a training dataset. A test utterance is then determined as belonging to the same class as the template with the lowest distance. In ASR, this distance is typically based on dynamic time warping (DTW), which deals with the variable length of the sequences of speech feature vectors representing the templates and the test utterances. Parametric models for ASR are based on HMMs (Rabiner, 1989), where an underlying first order Markov process involving state-to-state transitions is assumed. This process is not directly observable, but can be inferred through the sequence of features extracted from the speech signal. The parameters of HMMs characterizing the state transitions and the state emission likelihoods are estimated by optimizing an objective function on a training dataset.

A critical issue in both TM and HMM-based approaches is the definition of the similarity measure required to classify the feature vectors extracted from a test utterance. In TM, this is represented by a local distance between the vectors forming the templates and the vectors from the test sequence. On the other hand, the similarity measure in HMMs is provided by the emission likelihood distribution assigned to each state. Traditional similarity measures are the Euclidean (or Mahalanobis) distances as local distances in TM and likelihoods calculated from Gaussian mixture models (GMMs) for estimating the state emission distributions in HMMs. These general-purpose measures have shown to be appropriate for standard speech features characterizing the speech spectrum.

More recently, posterior probabilities have been used as speech features (Hermansky *et al.*, 2000a). In this case, each feature vector is composed by the posteriors of sub-word classes given the spectral features. In this thesis, these posterior-based speech features are referred to as pos-

terior features. Unlike standard spectral-based feature vectors, the components of the posterior features hold some specific properties, including the fact that they are non-negative and sum up to one. However, applying the general-purpose similarity measures presented above (Euclidean distance and GMM-based likelihood) present some drawbacks. Aside from not taking into account the specific properties of posterior features, it is often necessary to employ "ad-hoc" transformations in order to use such measures with posterior features, thereby removing their probabilistic nature. Thus, the convenient characteristics of posterior features are no longer fully exploited.

1.3 Motivation for Using Posterior-based Features

The input of ASR systems consists of a parametric representation of the speech signal to be recognized. This speech representation involves a sequence of feature vectors characterizing the utterance. Traditional speech feature vectors describe the short-term spectrum of the speech signal based on models of speech production and perception. The purpose of feature extraction is mainly to reduce the dimensionality of the speech signal while preserving (or enhancing) the discriminant information of the data. In the process, the irrelevant variability should be reduced while the relevant variability should be preserved. Following this direction, posterior probabilities of sub-word units can be used as speech features. Although the present thesis will only focus on posterior classes representing phonemes[1], other sub-word classes can be used, including those resulting from unsupervised clustering. Compared to spectrum-based speech features, posterior estimates present the following potential advantages:

- Since ASR systems should be independent of the speaker and the environment, the speech signal to be recognized can be reduced to a "robust" (although still noisy) sequence of phonemes. In this case, phoneme posteriors are a very convenient set of speech features because they capture much of the phonetic information contained in the signal.

- State-of-the-art acoustic models represent phoneme-level units. Using a feature extractor that explicitly conveys information about the set of phonemes adds consistency between the feature extractor and the acoustic modeling.

[1]Phonemes can be defined as the unique sound categories that serve to distinguish between the meanings of different words (Gold and Morgan, 1999).

- Posterior probabilities are the detectors that minimize the error in a Bayesian classification framework (Duda *et al.*, 2001). Hence, speech features representing (phoneme) posteriors can be seen as the optimal (phonetic) representation.

- Posterior features are obtained from a transformation of the features characterizing the spectrum of the signal. This transformation maximizes the mutual information between the set of phonemes and the set of spectral-based features[2]. Hence, from the information theoretic point of view, posteriors are also the optimal representation of the phonetic structure of a spoken utterance.

The posterior feature estimator is learned from a training dataset. This contrasts with the extraction process of standard spectral-based features, which is based on a transformation mainly inspired from perceptual models. Using a training dataset for estimating the posterior features presents the following advantages:

- The feature extractor for estimating the posterior probabilities does not rely on perceptual models. Instead, it is based on data-driven methods that extract the speech information contained on a training dataset. The criteria for estimating these data-driven methods can then be specific to improve the ASR accuracy.

- The estimation of the posterior transformation can follow a discriminative procedure. Thus, the ability to distinguish between the different posterior classes is optimized. Although in theory, minimizing the classification error at the frame level does not necessarily implies a minimization of the word error rate, in practice, it yields to a better recognition performance (Shire, 2001).

- Extraction methods for standard spectral-based speech features assume that the input signal is stationary. Therefore, a short temporal context is used to compute each spectral-based feature vector. If the method to estimate the posterior probabilities does not make any assumption about the inputs, a longer temporal context can then be used. This property can be convenient for estimating phoneme posteriors since phonetic information can be spanned along a syllable-length temporal interval (Yang *et al.*, 1999).

[2]This relation is developed in Section 2.2.2.

- If the dataset used for training the estimator of the posterior probabilities is rich enough in term of speakers and environments, posteriors can then be considered as speaker and environment-invariant speech features.

1.4 Contribution of this Thesis

This thesis proposes a principled acoustic model for ASR that directly uses posterior probabilities of phonemes as input features. In the proposed framework, the Kullback-Leibler (KL) divergence (Kullback and Leibler, 1951) is used as similarity measure. Since this measure is a natural distance between posterior probabilities (Cover and Thomas, 1991), posterior-based speech features do not need to be transformed into another domain and moreover, their properties are explicitly considered. This paradigm is investigated in two different contexts:

- Template Matching (TM): The main limitation of traditional template-based approaches using spectral-based features is the huge amount of training data that is typically required to obtain a suitable characterization of the speech variability. Therefore, this approach has traditionally been applied in speaker-dependent and small vocabulary systems. This thesis explores the adequacy of using posterior features in a template-based framework when the KL divergence is used to compute the local distance. This method is compared to state-of-the-art acoustic models in different databases that cover different situations in terms of lexicon and training data sizes.

- Hidden Markov Models (HMMs): A parametric model is investigated to capture the speech variability of the posterior features from a training dataset. This model follows the same topology as typical HMMs used in ASR. The main difference relies on the definition of the state emission distribution. The KL divergence is used to compute the similarity between the posterior features and a reference multinomial distribution characterizing each state. This model establishes a general framework that generalizes and unifies other posterior-based acoustic models found in the literature. Moreover, the proposed model is interpreted using different criteria that range from maximum likelihood to information theoretic clustering.

1.5 Organization of this Thesis

The content of this thesis is structured as follows:

- Chapter 2 presents the main components of an ASR system. These components involve feature extraction, acoustic modeling, language modeling and decoding. Special emphasis is given to the feature extraction module. The two main methods (MFCC and PLP) for computing standard spectral-based features are described. Then, the most common technique for estimating the posterior probabilities of phonetic classes is presented. This procedure is based on a type of artificial neural network, the multi-layer perceptron (MLP). Algorithms for training this model and generating the posteriors estimates are discussed in depth. Finally, the different databases along with the evaluation method of the ASR approaches used in this thesis are described.

- Chapter 3 presents in detail the two main types of acoustic models for ASR. They correspond to a non-parametric technique based on templates and parametric models represented by HMMs. These models typically use standard spectral-based speech features as inputs. In this thesis, the acoustic models described in this chapter are extended in the following chapters so that they can benefit from the properties of the posterior features while preserving the advantages of state-of-the-art ASR systems.

- Chapter 4 presents a survey of state-of-the-art acoustic models for ASR that use posterior probabilities of sub-word units. The main goal of this chapter is to present the different roles that posterior probabilities can take in ASR. We characterize the different systems in this chapter according to the method used to estimate posteriors.

- Chapter 5 constitutes the first part of the main contribution of this thesis. Template matching (TM) approach is adapted so that properties of posterior features can be fully exploited. Different local distances based on the KL divergence are discussed. These local distances are compared to state-of-the-art ASR systems using different databases that simulate different conditions in terms of lexicon and training data sizes.

- Chapter 6 describes the second part of the contribution of this thesis. A parametric approach based on the HMM architecture is studied. This model directly uses posterior features as

inputs. The score associated to each state is derived from the KL divergence. This measure is further justified by interpreting the proposed model using different criteria ranging from maximum likelihood to information theoretic clustering. Furthermore, this model provides a general framework where other acoustic models using posterior probabilities as inputs are particular cases. The performance of the proposed model along with its complexity are compared to state-of-the-art acoustic models.

- Finally, Chapter 7 summarizes the contribution of this thesis in the framework of both TM and HMM contexts. The main conclusions of the application of the KL-based measures in acoustic models using posterior features as inputs are also exposed. This chapter is then concluded by presenting the future directions that can be taken for further improving the acoustic models proposed in this thesis.

Chapter 2

Fundamentals of Speech Recognition

2.1 Introduction

An ASR system follows the structure of a pattern classification task (Duda *et al.*, 2001). The speech signal is first processed to extract the characteristics that are necessary for the recognition of the linguistic message. Then, a distance score is computed between the speech features and each reference class and a classification decision is finally made according to the distance scores. As shown in Figure 2.1, these three steps are respectively implemented in an ASR system by 1) the feature extractor, 2) the acoustic model and 3) the decoder. This chapter describes these elements and also presents the databases that are used in this thesis.

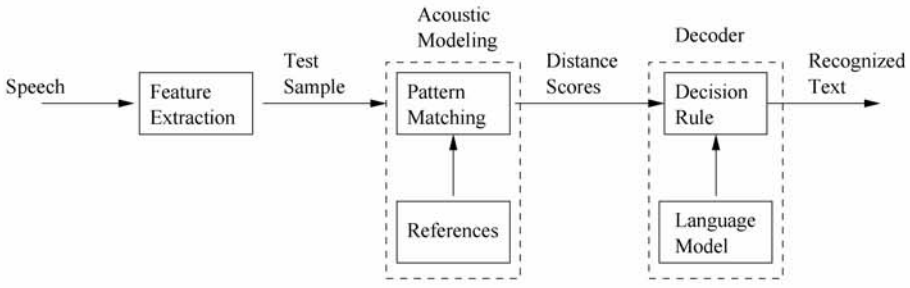

Figure 2.1. Block diagram of an ASR system

The input of an ASR system is a digital speech signal typically sampled at 8 or 16 KHz. This signal contains redundant or irrelevant information for the recognition task, such as characteristics of the speaker and the environment (Rabiner and Juang, 1993). The feature extractor component removes the information which is not related to the linguistic message W represented by the speech signal. Thus, characteristics related to the speaker or the environment are attempted to be discarded. A vector of features is computed from a fixed-length temporal window of the speech signal. Hence, a sequence of feature vectors is obtained from the speech signal through this module. Section 2.2 describes the two main types of feature extractors that are used in this thesis: cepstrum and posterior-based speech features. The latter is derived from a data-driven post-processing of the former method. The main work of this thesis is based on this latter type of feature extractor.

The ASR problem can be expressed within the Bayesian framework (Jelinek, 1976, 2001). Given a sequence of speech features $X = \{\mathbf{x}_1, \cdots, \mathbf{x}_t, \cdots, \mathbf{x}_T\}$, the most probable linguistic message (sequence of words) $\hat{W}$ is then formulated as

$$\begin{aligned} \hat{W} &= \arg\max_{W \in \mathcal{W}} P(W|X) && (2.1) \\ &= \arg\max_{W \in \mathcal{W}} \frac{p(X|W)P(W)}{p(X)} && (2.2) \\ &= \arg\max_{W \in \mathcal{W}} \log p(X|W) + \log P(W) && (2.3) \end{aligned}$$

where $\mathcal{W}$ is the set containing all the possible word sequences. Step (2.2) is obtained by applying Bayes' rule. In this step, the term $p(X)$ can be ignored because it does not affect the maximization solution. Also, we can observe that two independent terms are generated. The first term, $p(X|W)$, depends on the sequence of speech features X and is known as acoustic model. The second term, $P(W)$, corresponds to the language modeling and represents the prior knowledge about the sequence of words W. In practice, computations are done in the logarithm domain to avoid numerical instabilities as shown in step (2.3). Generally speaking, the term $\log p(X|W)$ can be interpreted as a similarity score $J^W(X)$ between the sequence X and the reference model corresponding to W.

Section 2.3 describes the main principle of the acoustic models applied in ASR. They are typically based on state sequences that are matched with the sequence of feature vectors X. The states of these sequences can be characterized by either parametric stochastic models or speech features. Chapter 3 explains in detail these two approaches for acoustic modeling. This thesis mainly focuses

on adapting state-of-the-art acoustic models to explicitly consider the properties of posterior-based speech features.

Section 2.4 briefly describes the main statistical techniques used by state-of-the-art ASR systems for estimating the term corresponding to the language model, $P(W)$. Section 2.5 presents the criteria and the algorithms to efficiently search on the space of the word sequences $\mathcal{W}$ as presented in (2.3).

Finally, Section 2.6 describes the databases used in this thesis. They correspond to different applications and the acoustic models studied are thus applied according to the characteristics of each database.

2.2 Feature Extraction

As described in the introduction of this chapter, the purpose of feature extraction is to obtain the characteristics of the speech signal that are relevant for recognizing the linguistic message W. Hence, these features should represent the sounds describing the message and ignore information about the speaker or the environment. The characterization of these sounds can be classified as *low-level* or *high-level* depending on the complexity of the extracted information (O'Shaugnessy, 2003). The former case refers to those features which simply describe the speech spectrum. They are typically obtained by a mathematical transformation of the speech signal. High-level features, on the other hand, provide a more compact representation because they classify the speech sounds into some acoustic or phonological classes. Although the latter type of features are potentially more effective for ASR because they provide a finer description of the message W, they are more complex to estimate and less reliable as their estimation is based on a non-linear classification.

In Section 2.2.1, we present the two most common methods to estimate low-level speech features. They provide a parametric representation of the short-time speech spectrum and hence, they correspond to low-level characteristics of the speech signal. Their processing steps emulate some properties of the human auditory and speech production. On the other hand, the feature extractor discussed in Section 2.2.2 is more related to high-level speech characteristics. They represent a classification over a set of linguistic classes and they are obtained through a data-driven transformation of the spectral-based features. This type of features present some advantages over the

low-level speech features that are also described.

2.2.1 Cepstrum-based Speech Features

Speech is the result of the air flow generated by the lungs and modulated by a time-varying channel characterizing the vocal tract. The speech signal $s(t)$ can then be mathematically expressed as the convolution between an excitation $e(t)$ and the filter describing the vocal tract $v(t)$.

$$s(t) = e(t) * v(t) \tag{2.4}$$

The excitation $e(t)$ provides speech information like voicing, amplitude and pitch frequency, which are more influenced by the semantic context than by the individual sounds being produced. The spectral envelope $|V(w)|$ embodies the vocal tract resonances referred to as formants, of which the location and bandwidth are more representative of the sound (phoneme) being produced. Thus, cepstrum-based speech features parameterize the spectral envelope in the form of a 8 to 14 dimensional feature vector.

For speech acquisition, the acoustic waves are captured by a system that is band limited in the frequency domain. This frequency limitation determines the sampling rate of the speech signal. When the speech is collected over a telephone channel, the speech signal is sampled at 8 KHz. When using a microphone, the speech signal is typically sampled at 16 KHz. The databases used in this work contain speech utterances recorded in both cases, telephone and microphone channels.

The spectral processing explained in this section assumes that the signal is stationary. Since speech is not a stationary signal because it is produced by a time-varying system, features are extracted from windowed version of the signal. This window is typically chosen to be the Hamming window because its energy leakage in the secondary lobes is lower than rectangular window. The length of this time window is short enough (typically around 25 ms) so that speech signal within this window can be considered quasi-stationary. Also, in order to guarantee that all the speech samples are equally considered, the time windows are overlapped roughly 15 ms. Hence, a vector of speech features is generated every roughly 10 ms.

This section describes the two most common feature extractor methods used in state-of-the-art ASR systems: mel frequency cepstral coefficients (MFCCs) (Davis and Mermelstein, 1980) and

perceptual linear prediction (PLP) cepstral coefficients (Hermansky, 1990). Their purpose is similar for both of them: to obtain a parametric representation of the vocal tract filter $v(t)$. MFCC features achieve that by selecting the lower coefficients of a Fourier representation whereas PLP processing performs a low-order autoregressive model (Makhoul, 1975). The main steps to estimate MFCC and PLP features are (Gold and Morgan, 1999):

1. Estimation of the power spectrum of the speech signal within each time window. In the case of MFCC extraction, the log power spectrum is estimated. This transformation to the frequency domain is due to the fact that the spectral information of the speech signal is much more informative for speech sound discrimination than the time domain signal (Rabiner and Juang, 1993).

2. The energy at each frequency band is computed. These frequency bands are chosen such that their sensibility (width) is inversely proportional to their frequency center. This is supported by perceptual experiments showing that the human ear is more sensitive to lower than higher frequency tones. In the case of MFCC, the filter banks are triangular and the frequency bands are based on the speech perception (mel scale). This mel scale corresponds to a linear spacing up to 0.5 KHz and thereafter a logarithmic spacing (Moore, 1997). It follows the spatial relationship of the hair cell distribution in the cochlea of the inner ear. In the case of PLP, trapezoidal filters are applied at roughly 1-Bark intervals. The Bark axis is derived by using the warping function (Hermansky, 1990)

$$\Omega(w) = 6 \log \left\{ \frac{w}{1200\pi} + \left[\left(\frac{w}{1200\pi} \right)^2 + 1 \right]^{0.5} \right\}$$

3. Pre-emphasis of the spectrum to approximate the unequal sensitivity of the human ear at different frequencies. The energy at low frequencies is higher than at high frequencies. Consequently, low frequency components are better represented than high frequency components. In order to equalize this spectral imbalance, a weighting is applied to the critical band energies in the case of PLP. In the case of MFCC, a filter that enhances the higher frequency terms of the speech signal is applied before computing the first step. The filter has the structure of $s(t) - as(t-1)$ where a is typically 0.95. This filter flattens the spectrum by increasing the magnitude of the high frequency components. This step is not relevant when using the

MEL scale since this scale is already amplifying high frequencies. For the PLP processing, an additional operation that compresses the spectral amplitudes is performed. This operation emulates the power-law relationship between intensity and loudness (Stevens, 1957). This is approximated by taking the cubic root of the critical band energies.

4. A smoothing of the spectrum is performed in this step by removing the high frequency components. In the case of MFCC, this step is done through a truncation of the coefficients of the inverse discrete Fourier transform (IDFT). Typically, the lower 13 coefficients are computed (12 coefficients plus the log-energy). In the case of PLP, a low-order autoregressive model (Makhoul, 1975) is applied. The purpose of this step is to remove the excitation $e(t)$ from the speech signal $s(t)$ based on the fact that it contains higher spectral fluctuations than $v(t)$. In this work, we use an autoregressive model of order 8, which is a typical value used in ASR.

5. The final step consists of an orthogonal representation of the features. In the case of MFCC, orthogonalization is already done while performing the IDFT. In the case of PLP, the coefficients of the autoregressive models are converted to cepstral coefficients through a simple recursion (Rabiner and Schafer, 1978). This orthogonal representation is useful when computing distances because it reduces the amount of parameters to estimate when modeling these features and also it reduces the time for computing the score of the acoustic models. For example, Mahalanobis distance or Gaussian distribution only require a diagonal covariance matrix.

6. An additional step is usually done to equalize the variances of the cepstrum coefficients. Each component is weighted by its index $c'(n) = nc(n)$. This step is not relevant when using a GMM for modeling the features since GMM directly scales the data through the covariance matrix.

The estimation of these speech features is done independently for each time window. However, there is a strong correlation between adjacent frames due to the continuous nature of the speech signal and the overlapping shifting between time windows. In order to take into account the time correlation, the first and the second order local temporal derivatives (Furui, 1986) (known as delta and double delta features respectively) are concatenated to the vector of speech feature. These derivatives are estimated as the slope of a linear regression among a context of typically 5 frames.

Hence, typical speech features contain 39 dimensions (13 static features + 13 delta features + 13 double delta features).

The ASR problem would be greatly simplified if the speech features corresponding to the same sound had similar values. Although the speech processing described above reduces the information related to the speaker and environment, spectral-based speech features still suffer from a high variance in the feature space of a sound. The next section describes a method to transform these spectrum-based features into a more stable speech representation. This transformation is based on data-driven techniques.

2.2.2 Posterior-based Speech Features

Unlike the speech features presented in the previous section, the feature extraction method discussed in this section attempts to provide high-level information of the speech signal. In particular, cepstral-based feature vectors are used as input of a classifier. The outputs of the classifier are estimates of the posterior probabilities of the target classes given the input. Since in this work, we use phonemes as classes, posterior features are estimates of phonemes posterior probabilities given the cepstral-based features.

The most common method to estimate posterior probabilities of sub-word units, such as phonemes is through a MLP because it scales well with large amount of training data and it can easily incorporate contextual information. However, other methods ranging from maximum entropy-based models (Hifny and Renals, 2005) to support vector machines (Salomon *et al.*, 2002) can also be found in the literature. Chapter 4 presents an overview of these estimation methods.

General Structure of the MLP

A MLP has a layered architecture consisting of an input layer, one or more hidden layers and an output layer. In this work, we use one-hidden-layered MLPs. The values of the hidden and the output layers are computed using a non-linear function of the values of the nodes of the previous layer. This structure is illustrated in Figure 2.2. Biases are used at the hidden and output layer to model those classification boundaries that do not contain the origin of the parameter space. The non-linear functions are defined by a set of weights which are estimated from a training dataset given an error criterion. Let w_{ij}^h be the weight connecting the input node i with the hidden node j

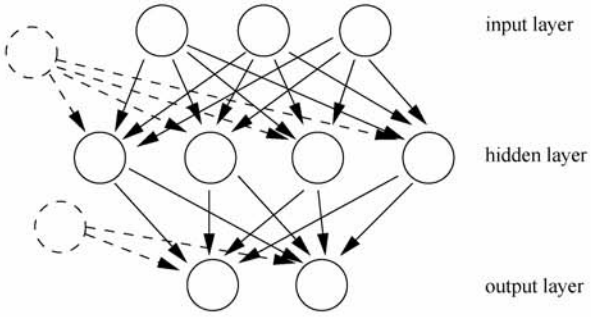

Figure 2.2. Structure of a MLP with one hidden layer. Dashed circles and arrows refer to biases.

and let v_j^i be the value for input node j. Then, the value of the hidden unit v_j^h is defined as

$$v_j^h = f_h(\{v_k^i\}_{k=1}^{N_i}) = \frac{1}{1 + \exp\left(\sum_{k=1}^{N_i} w_{kj}^h v_k^i + b_j^h\right)} \tag{2.5}$$

where N_i is the number of input units and b_j^h denotes the biases of the hidden layer. Since the number of hidden units N_h is usually much higher than the number of inputs, the hidden layer serves to map the inputs onto a higher non-linear space where class boundaries can be easier to obtain (Cover, 1965). In the case of the output layer, let us define w_{jk}^o the weight between the hidden unit j and the node k. The value v_k^o of the output node k is defined using the softmax function

$$v_j^o = f_o(\{v_k^h\}_{k=1}^{N_h}) = \frac{\exp\left(\sum_{k=1}^{N_h} w_{kj}^o v_k^h + b_j^o\right)}{\sum_{j'=1}^{N_o} \exp\left(\sum_{k'=1}^{N_h} w_{k'j'}^o v_{k'}^h + b_{j'}^o\right)} \tag{2.6}$$

where N_o is the number of output units and b_j^o denotes the biases of the output layer. This function is a continuous and differentiable approximation of the step function, which is normalized to guarantee that the sum of the output values is one. The step function can be seen as a threshold-based binary classifier of the values in the hidden layer $\{b_j\}_{j=1}^{N_h}$ (Bridle, 1989). The threshold and the slope of this modified step function are determined by the weights $\{w_{kj}^o\}$.

Estimation of the MLP Parameters

The parameters of the MLP, Θ_{MLP}, are the weights that define the non-linear functions that determine the values at the hidden and output layers, $\Theta_{MLP} = \left(\{w_{ij}^h\}, \{w_{jk}^o\}, \{b_j^h\}, \{b_j^o\}\right)$. Continuity and differentiability are necessary properties of the global function described by the MLP $\{c_k\}_{k=1}^{N_o} = (f_o \circ f_h)(\{x_i\})$ because the training procedure is based on the iterative gradient descent method (Rumelhart *et al.*, 1986). The error criterion is typically based on minimum squared error E_{MSE} ($\sum_{k=1}^{N_o} ||t_k - o_k||^2$) or relative entropy E_{RE} ($\sum_{k=1}^{N_o} = t_k \log o_k$) between the output values $\{o_k\}$ and the targets $\{t_k\}$.

There are typically two strategies to estimate the parameters of the MLP: *batch* or *on-line*. In the batch mode, weights are updated after obtaining the statistics from all the training data. In the on-line mode, weights are updated after every training sample. Though convergence using the on-line criterion is based on the existence of an infinite amount of training samples (Amari, 1967; Fu, 1968), in practice both modes yield to similar results. The criterion for stopping the iterations of the gradient descent method is usually based on the accuracy on a cross-validation set. This prevents the MLP to be over-trained. In this work, MLPs are trained using the relative entropy criterion and the on-line mode because convergence is faster.

The MLP as a Posterior Estimator

Given a large enough number of hidden units, it can be proved that a MLP can estimate any multi-variate function (White, 1990). In particular, in (Richard and Lippmann, 1991; Bourlard and Wellekens, 1990) it is shown that a MLP with inputs $I = \{v_j^i\}$ and desired target outputs $T = \{t_k\}$ estimate the conditional expectation $E[T|I]$. If the target outputs T are discrete with value one for a given class c_k and zero for the rest of classes, then the expectation becomes $E[T = c_k|I] = P(c_k|I)$. This holds for both training criteria E_{MSE} and E_{RE}. In this thesis, the classes c_k represent phonemes and a context of cepstrum-based speech features is used as input[1]. Hence, the output values of the MLP are estimates of the phoneme posterior probabilities given the low-level speech features described in Section 2.2.1. The posterior-based speech feature is then a vector formed by output values at each time frame. At the time frame t, a left and right context of typically four frames, $(\mathbf{x}_{t-4}, \cdots, \mathbf{x}_t, \cdots, \mathbf{x}_{t+4})$, is used as input for the MLP. Then, the output values represent

[1]The input features are normalized in mean and variance to improve the convergence speed (Joost and Schiffmann, 1998).

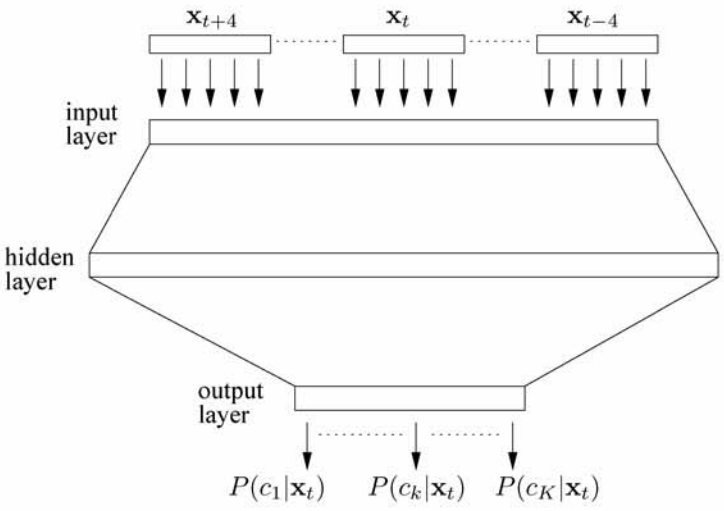

Figure 2.3. Structure of the MLP as a posterior estimator. A context of adjacent cepstral features is used as input. The set of output values $\{P(c_k|\mathbf{x}_t)\}$ forms a vector called posterior feature frame.

the posterior probability of the phoneme c_k given the input, i.e. each output is an estimate of $[P(c_k|\mathbf{x}_{t-4}, \cdots, \mathbf{x}_{t+4})]^{\mathsf{T}}$. The posterior feature vector $\mathbf{z}_t$ is then formed by the set of MLP output values[2] $\{P(c_1|\mathbf{x}_t), \cdots, P(c_K|\mathbf{x}_t)\}$ where K represents the total number of phonemes and also, the number of output nodes, thus $N_o = K$. This structure is represented in Figure 2.3.

When compared to cepstrum-based speech features, posterior features present the following advantages:

- The MLP is trained to minimized the phoneme error at the frame level. Though minimizing the phoneme error rate does not guarantee the minimization of word error rate, in practice, both errors are generally correlated (Shire, 2001).

- They are discriminant features since the MLP is trained using a discriminative criterion.

- The non-linear functions that characterize the MLP nodes can segment the feature space using non-linear boundaries.

- Since a context of typically 9 frames is used as input of the MLP, posterior features use a longer context of approximately 100 ms.

[2]For the sake of simplicity, in the rest of this thesis we express the posterior probability conditioned to only one frame, i.e., $P(c_k|\mathbf{x}_t)$.

- If the MLP is trained on a rich enough database, in terms of speakers and vocabulary, posterior probabilities can be considered speaker-independent features.

- The MLP can accept a wide range of features as input without making any assumption about their distribution. For instance, the gender information (Konig and Morgan, 1992) or the pitch frequency (Doss *et al.*, 2003) can be fed into the MLP along with the standard continuous features.

- Each component of the feature vector corresponds to a specific phoneme and contains a linguistically meaningful value.

- Since posterior features can be seen as discrete distributions, measures from the information theory field can be applied. For example, the entropy can be computed to measure the uncertainty of of the MLP classification output and the Kullback-Leibler (KL) divergence can be used to measure the similarity between two posterior vectors.

- The output of different MLPs trained on different training data can be combined because all the outputs estimate a posterior-based features. Several combination strategies between posteriors have been explored in the literature (Misra *et al.*, 2003; Valente and Hermansky, 2007).

- As mentioned in the introduction, the transformation of cepstral-based features into posterior features maximizes the mutual information $I(\mathcal{C}, \mathcal{X})$ between the set of cepstral-based features $\mathcal{X}$ and the set of MLP classes $\mathcal{C}$ (phonemes in this work). This can be shown using the relation $I(\mathcal{C}, \mathcal{X}) = H(\mathcal{C}) - H(\mathcal{C}|\mathcal{X})$, where the entropy of the classes, $H(\mathcal{C})$, is constant. Hence,

$$\begin{aligned} \max I(\mathcal{C}, \mathcal{X}) &= \min H(\mathcal{C}|\mathcal{X}) && (2.7) \\ &= -\min \sum_{c} \int_{\mathcal{X}} p(c, \mathbf{x}) \log P(c|\mathbf{x}) d\mathbf{x} && (2.8) \\ &\approx -\min \frac{1}{N} \sum_{n} \log P(c|\mathbf{x}_n) && (2.9) \end{aligned}$$

  where the sum on the last step is performed over all samples used for training the MLP. The last expression corresponds to the relative entropy criterion used for estimating the MLP weights.

This thesis focuses on investigating novel acoustic models that use posterior feature vectors directly as input features. These models explicitly consider the particular properties of posteriors through an appropriate similarity measure.

2.3 Acoustic Modeling

In the previous section, we describe how to extract the features X from the speech signal that better characterize the underlying linguistic message W. Each possible message W in the space of linguistic messages $\mathcal{W}$ is represented by a reference acoustic model. The acoustic model score $J^W(X)$ which yields a high ASR accuracy should be high for those sequences X that represent the message W and low for the rest.

Ideally, two sequences of feature vectors should be the same if and only if they contain information about the same linguistic content. However, the variability inherent in the speech signal is still present in the sequence of extracted feature vectors X. In particular, the acoustic model must deal with two types of variability:

- Time variability: As described in the previous section, a vector of speech features is extracted roughly every 10 ms of speech signal. Consequently, the number of feature vectors is linearly proportional to the duration of the speech utterance. The same linguistic message can be pronounced using different speech rates. Thus, the length of vector sequence differs from one pronunciation to another.

- Feature variability: Ideally, the feature extractor module should provide a unique representation for each different sound. However, speech features X exhibit considerable variability. This variability leads to a high variance within the feature space of a phoneme. The acoustic model must then be able to characterize the region on the feature space corresponding to each phoneme.

To deal with these two types of variability, the main principle of the most common acoustic models used for ASR is to represent each linguistic message W with a sequence of states which is warped so that each feature vector $\mathbf{x}_t$ is associated to a state $q_t = i$. The score of the acoustic model is then based on the difference between the sequence of speech vectors $X = \{\mathbf{x}_1, \cdots, \mathbf{x}_T\}$ and the

warped state sequence $\{q_1, \cdots, q_T\}$. In this way, both time and feature variability can be modeled. Time variability is modeled by the warping map that associates feature vectors with states and feature variability is represented by the difference between the warped state sequence and X.

Each state i characterizes a region of the feature space. This characterization can be done with a parametric probability distribution or with a distortion measure. In the first case, the parameters of the probability distribution are required. Hence, the probability distribution can be written as $p(\mathbf{x}_t|q_t = i, \Theta_a)$ where Θ_a denotes the set of parameters. This distribution estimates the probability that a feature vector $\mathbf{x}_t$ is generated by the region associated to the state i. In this case, the region of the acoustic space typically corresponds to phoneme-level units. The set of parameters Θ_a is estimated from a training dataset. This approach is based on hidden Markov models (HMMs) and is discussed in detail in Section 3.2. When the region is characterized by a distortion measure, the sequence of states is a sequence of speech features corresponding to a linguistic unit. This acoustic sample is known as template. The distortion measure is then computed between the sequence of speech features extracted from the test utterance and the sequence of speech features representing the template. This method is referred to as template matching (TM) and is fully described in Section 3.3. Although HMM can obtain a richer characterization of the acoustic regions on the feature space, TM can better model the dynamics of the trajectory on the feature space defined by the speech features.

In this work, the Kullback-Leibler (KL) divergence is used as similarity measure in both TM and HMM-based acoustic models for posterior features. This measure explicitly considers the properties of this type of features.

2.4 Language Modeling

The most likely sequence of recognized words W is chosen depending on the score computed from the acoustic model $p(X|W)$ and the prior probability of the sequence of words $P(W)$ as expressed in (2.3). This latter term is estimated through the language model.

The probability of a sequence of L words $W = \{w_1, \cdots, w_L\}$ can be mathematically decomposed

as

$$P(W) = \prod_{l=1}^{L} P(w_l|w_1, \cdots, w_{l-1}) \tag{2.10}$$

This formulation reflects the idea that the probability of each word in the sequence W is conditioned to the precedent words. Since the product terms are difficult to estimate, they are approximated by using only the N previous words. This corresponds to a Nth order Markov chain and is referred to as N-gram model.

$$P(W|\Theta_l) \approx \prod_{l=1}^{L} P(w_l|w_{l-N}, \cdots, w_{l-1}) \tag{2.11}$$

where Θ_l is the set of estimates of the conditional probabilities $P(w_l|w_{l-N}, \cdots, w_{l-1})$. These parameters are typically estimated from the word relative frequencies in a training dataset

$$P(w_l|w_{l-N}, \cdots, w_{l-1}) = \frac{N(w_{l-N}, \cdots, w_l)}{N(w_{l-N}, \cdots, w_{l-1})} \tag{2.12}$$

where $N(\cdot)$ represents the number of times that a particular event occurs. In practice, the number of L words that can be combined increases exponentially with N and also, the amount of training data required to estimate the probabilities. Smoothing techniques are usually employed to assign some non-zero probability to those word combinations unseen in the training data. The most common method is known as "back-off" and is based on interpolating lower order statistics (Katz, 1987). The type of language model depends on the recognition task: unigram (N = 0) is used when all words are equally probable, for example, recognition of digit sequences. Models using $N = 1$ (bigram) or $N = 2$ (trigram) are generally used in large vocabulary continuous speech recognition tasks.

2.5 Decoding

Decoding is the last component of an ASR system. It uses the information from the acoustic model and the language model to obtain the most likely sequence of words $\hat{W}$.

$$\hat{W} = \arg\max_{W \in \mathcal{W}} J^W(X) + \log P(W|\Theta_l) \tag{2.13}$$

The acoustic score $J_W(X)$ and the language model $P(W|\Theta_l)$ must be evaluated for each possible message W. This search can be done efficiently by a form of dynamic programming (Bellman, 1966).

In order to further reduce the computational time for large vocabulary systems, those hypotheses with a partial score lower than a given threshold are discarded. This technique is known as beam-search (Lee *et al.*, 1990). This pruning method can be also applied only at the end of each word. Although such approach makes the search faster, it can also discard some correct hypotheses prematurely.

The acoustic score and the language model probabilities have different dynamic ranges. While the score from the acoustic model is very low because it is the joint likelihood of T (number of frames) transition probabilities and T emission likelihoods, the language model probabilities are simply the joint probability of L (number of words) conditional probabilities. Hence, the score provided by the acoustic model may dominate the score from the language model. In order to overcome this problem, the logarithm of the language model score is scaled. This language model scale factor is empirically determined to maximize the ASR accuracy.

During recognition, most of the errors come from the insertion of short words. This is due to a relatively high acoustic model likelihood and the wide range of contexts where these words can appear. To alleviate this problem, a penalty factor is inserted between words to penalize those sequences containing a large number of words.

2.6 Databases

In this thesis, we have studied different tasks, corresponding to different amount of training data and lexicon sizes.

2.6.1 Phonebook

We use the Phonebook speech corpus for speaker-independent task-independent, small vocabulary (75 words) isolated word recognition (Pitrelli *et al.*, 1995) over a microphone channel (16KHz). In this work, we use only the test set. The definition of the test set if obtained from (Dupont *et al.*, 1997). It consists of 8 different sub-sets of 75 different words each set. Each word is pronounced a dozen times by different speakers. There are 45 different phonemes including silence. The acoustic

	Digits	RM	WSJ	CTS
inputs	351	351	351	351
hidden units	1000	1500	3652	10000
outputs	27	45	45	45
training set	4h	3.8h	7h	225h
validation set	0.5h	1.4h	1h	35h
frame accuracy - training	83.9	80.9	83.9	64.9
frame accuracy - validation	82.0	73.2	81.4	63.6

Table 2.1. Characterization of the MLPs used in this work. The training and validation sets are expressed in hours. The frame accuracy expresses the percentage of frames that are correctly classified. A frame is correctly classified if its highest output class corresponds to the target class.

vector comprises PLP cepstral coefficients (Hermansky, 1990) extracted from the speech signal using a window of 25 ms with a shift of 10 ms, followed by cepstral mean subtraction. At each time frame, 13 PLP cepstral coefficients, their first-order and second-order derivatives are extracted (Furui, 1986), resulting in a 39 dimensional cepstral vector.

2.6.2 Digits Task

This database is a subset of the OGI-Numbers database containing spoken sequences of continuous numbers over the telephone channel (Cole *et al.*, 1995) (8KHz). Utterances formed by digits (from *zero* to *nine* plus *oh*) are selected, hence, the lexicon size is 11 words plus silence with a single pronunciation for each word. The training set contains 8253 utterances (approximately 4.5 hours) spoken by different speakers and the test set consists of 2820 utterances (Mariethoz and Bengio, 2004). The acoustic vector comprises PLP cepstral coefficients extracted from the speech signal using a window of 25 ms with a shift of 10 ms, followed by cepstral mean subtraction. At each time frame, 13 PLP cepstral coefficients, their first-order and second-order derivatives are extracted, resulting in a 39 dimensional acoustic vector. There are 27 context-independent phonemes including silence. A MLP has been trained from this database. Information about the MLP structure is shown in Table 2.1.

2.6.3 Resource Management (RM)

The Resource Management (RM) corpus consists of read queries on the status of Naval resources (Price *et al.*, 1988). The training set consists of 2880 utterances spoken by 109 speakers corresponding to approximately 3.8 hours of speech. The test set contains 1200 utterances amount-

ing to 1.1 hours in total. The test set is completely covered by a word pair grammar included in the task specification which is used for recognition. The lexicon is formed by 992 different words. Some functional words, such as the articles *a* or *the*, use multiple pronunciations. There are 45 phonemes including silence. The feature vector comprises PLP cepstral coefficients, their deltas and delta-deltas (39 dimensions) using a window of 25 ms with a 10 ms frame shift. Details of the MLP trained on this database are shown in Table 2.1.

2.6.4 Wall Street Journal (WSJ)

This database is formed by sentences from the Wall Street Journal newspaper (Paul and Baker, 1992). Sentences are read by multiple speakers and recorded on microphone channel (16 KHz). The training set consist of 38250 utterances corresponding to roughly 80 hours. The test set is known as *si_dt_05*. The phonetic lexicon of the test data is formed by 913 utterances containing 4988 different words. Around 8% of the test set corresponds to words not appearing in this test lexicon (out-of-vocabulary words). Hence, perfect recognition can not be obtained. The language model is a bigram provided by the database. There are 45 phonemes including silence. The feature vector comprises PLP cepstral coefficients, their deltas and delta-deltas (39 dimensions) using a window of 25 ms with a 10 ms frame shift. Table 2.1 shows the information of the MLP trained on this database.

2.6.5 Conversation Telephone Speech (CTS)

This database is formed by continuous speech utterances from different speakers over a telephone channel (Godfrey *et al.*, 1992) (8 KHz). Only the training set of this database is used in this work. It contains 260 hours of gender balanced speech randomly selected from the Fisher Corpus and the Switchboard Corpus. This database is only used for estimating the parameters of a MLP. Information about the MLP structure can be found in table 2.1.

2.7 Evaluation of ASR Systems

In ASR research, the evaluation of speech recognition systems is performed for two major reasons: (1) to assess the performance of the speech recognition system and (2) to compare two different

recognition systems.

Generally, the speech recognition systems are evaluated on an unseen test set (data not used during training) in terms of word error rate (WER). The output of the speech recognition system is a sequence of words (automatic transcription). Given the reference sequence of words (ground truth), the evaluation of ASR system is done by comparing the two sequence of strings. This is usually done by computing the Levenshtein distance or the edit distance. The Levenshtein distance between two strings is the minimum number of changes that has to be made in one string to transform it into another (Sankoff and Kruskal, 1999). The changes are basically insertion, deletion, or substitution. The WER is then the Levenshtein distance or edit distance between the ground truth and automatic transcription normalized by the length of ground truth, i.e., if N_r is the number of words in the ground truth and the number of insertion, deletion, and substitution are I, D and S, respectively then the WER is estimated as (in terms of percentage)

$$WER = \frac{I + D + S}{N_r} 100 \tag{2.14}$$

In case of isolated word recognition task, there are no insertion or deletion errors, but only substitution errors. In this thesis, we evaluate the ASR systems in terms of WER. Also, the word insertion penalty is chosen so that the number of insertion and deletion errors is equivalent.

The performance of different speech recognition systems can be compared using a significance test. In this thesis, we use a statistical test based on the bootstrap estimate[3] (Bisani and Ney, 2004).

2.8 Summary

In this chapter, we described the different components of an automatic speech recognition system. This is also sketched in Figure 2.1. Special emphasis has been given to the estimation of posterior-based features because they constitute the central theme of this thesis. We also described the different databases used in this thesis along with the method to evaluate the ASR systems.

In the following chapter, we describe in detail the acoustic modeling process in template and

[3]Bootstrap is a method to determine the trustworthiness of a statistics. This is done by creating a replica of the statistics by random sampling from the data set with replacement (Efron and Tibshirani, 1993).

HMM-based ASR system. We also introduce the different HMM-based approaches that are used in the present work.

Chapter 3

Acoustic Models for ASR

3.1 Introduction

The main goal of acoustic modeling is to obtain a representation of each possible linguistic message W that matches the sequence of feature parameters $X = \{\mathbf{x}_1, \cdots, \mathbf{x}_t, \cdots, \mathbf{x}_T\}$ extracted from speech signals representing that message W. The reference representation of W is defined as a sequence of N^W states $\{1, \cdots, i, \cdots, N^W\}$ where each state i corresponds to a region on the space of speech features. This region usually corresponds to a sub-word unit, such as context-dependent phonemes. At each state i, an acoustic similarity measure is assigned to compute the matching between a feature vector $\mathbf{x}_t$ and the region associated to that state. The reference sequence is then warped to the length of X in such a way that the global matching between the speech features and the regions given by reference models of W is maximized. The score of the acoustic model $J^W(X)$ is thus equal to this global matching.

The comparison between the sequence of features X and a warped version of the reference sequence of W can deal with the both types of speech variability (time and feature) as discussed earlier in Section 2.3. The temporal variability is handled by the property of the state sequence of W to be warped to the length of X whereas the feature variability is modeled by the similarity function assigned to each state i. An example of this scheme is illustrated in Figure 3.1. In this example, the state sequence $\{1, 2, 3, 4\}$ is warped to a 6-element sequence $\{1, 2, 2, 3, 3, 4\}$ since the test sequence X contains 6 elements. This is the warping sequence that maximizes the matching

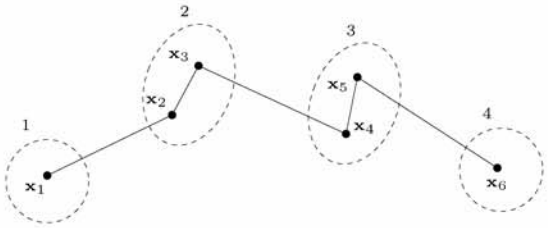

Figure 3.1. Each element of the feature sequence $X = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6\}$ is associated with one state of the sequence $\{1, 2, 3, 4\}$, where each state characterizes an acoustic region on the feature space.

with the sequence X.

In this chapter, we describe the two main approaches used in ASR to estimate the acoustic modeling score: hidden Markov models (HMMs) in Section 3.2 and template matching (TM) in Section 3.3. The main difference between these two methods lies on the characterization of the states. In HMMs, the matching function of each state $q_t = i$ is a probability distribution $p(\mathbf{x}_t|q_t = i)$ that computes the likelihood of the feature vector $\mathbf{x}_t$ being emitted by state i. The parameters of each state emission distribution are estimated from a training dataset to capture the variability present in the acoustic region represented by the state i. In this chapter, we also discuss the most common approaches to model the state emission probability distributions.

In the case of the TM approach, each state i is a feature vector $\mathbf{y}_i^W$. A similarity measure $\phi(\mathbf{y}_i^W, \mathbf{x}_t)$ is then defined to compute the matching between the speech feature $\mathbf{x}_t$ and the vector $\mathbf{y}_i^W$. The sequence of states $\{\mathbf{y}_1^W, \cdots, \mathbf{y}_i^W, \cdots, \mathbf{y}_{N^W}^W\}$ is a sequence of speech features obtained from an utterance in the training dataset corresponding to the linguistic message W. This sequence is referred to as template.

States from a HMM can better represent the speech variability than states from a template because (a) the parameters of the HMM states are estimated from a training dataset and (b) the emission likelihood distribution can describe more complex regions than a similarity function. However, TM can better characterize the temporal evolution of the speech features because templates correspond to real utterances. TM-based approaches attempt to obtain an accurate description of the speech variability by using several templates for each word. On the other hand, HMMs typically use only one reference model per word.

In the rest of this chapter, we first describe the architecture and the different procedures for training and decoding HMMs. The most common methods to estimate the state emission likelihoods

are presented. Then, the algorithms involved in TM are discussed, also identifying their links with HMMs.

3.2 Hidden Markov Models

3.2.1 General Description

The description of HMMs can be done from the Bayesian formulation of the ASR problem expressed in (2.2). The term corresponding to the acoustic model, $p(X|W)$, can be decomposed using an auxiliary variable Q in the following way

$$p(X|W) = \sum_{Q \in \mathcal{Q}(W)} p(X|Q,W)P(Q|W) \tag{3.1}$$

where $\mathcal{Q}(W)$ denotes the set of all possible state sequences $\{q_1, \cdots, q_t, \cdots, q_T\}$ of the same length T as the sequence of speech features $X = \{\mathbf{x}_1, \cdots, \mathbf{x}_T\}$. Each state q_t is a discrete random variable belonging to the set $[1, N^W]$, where N^W denotes the total number of states. As mentioned previously, each state defines a region on the space of speech features. Expression (3.1) can be reformulated as

$$p(X|W) = \sum_{Q \in \mathcal{Q}^W} p(X|Q)P(Q) \tag{3.2}$$

where $\mathcal{Q}^W$ denotes the set of all possible state sequences allowed by the linguistic message W, i.e., $\mathcal{Q}^W = \{Q \in \mathcal{Q} | P(Q|W) \neq 0\}$.

HMMs make two assumptions about the distribution of X:

- The vectors contained in X are assumed to be independent of each other given the state.

$$p(X|Q) = \prod_{t=1}^{T} p(\mathbf{x}_t|q_t) \tag{3.3}$$

  This condition assumes that the likelihood of a feature vector only depends on the state it has been emitted from. Other approaches like segmental HMMs have attempted to explicitly take into account the adjacent vectors when computing the emission likelihood (Ostendorf *et al.*, 1996).

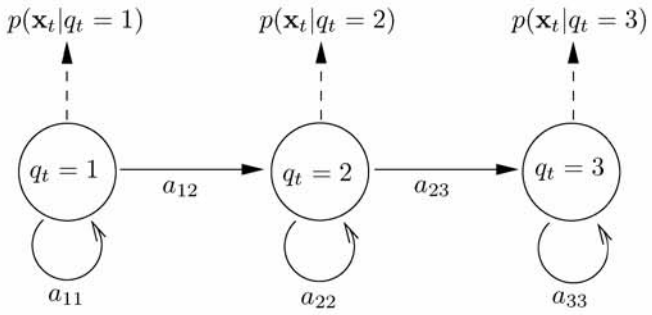

Figure 3.2. Structure of a left-to-right HMM formed by three states. Each state i is defined by an emission probability $p(\mathbf{x}_t|q_t = i)$. Transitions between states are described by the probability $a_{ij} = p(q_t = j|q_{t-1} = i)$.

- The distribution of Q follows a first order Markov chain.

$$P(Q) = \prod_{t=1}^{T} P(q_t|q_{t-1}) \tag{3.4}$$

  Hence, the probability of being in one state (region) at time frame t only depends on the state at the previous frame q_{t-1}. In ASR, HMMs typically have a left-to-right topology[1]. This means that state transitions only allow staying on the same state or moving to the next one.

Applying the above assumptions to (3.2), we obtain that

$$p(X|W) = \sum_{Q \in \mathcal{Q}^W} p(X|Q)P(Q) \approx \sum_{Q \in \mathcal{Q}^W} \prod_{t=1}^{T} p(\mathbf{x}_t|q_t)P(q_t|q_{t-1}) \tag{3.5}$$

We can then observe that HMMs are defined by two stochastic processes which actually correspond to the two types of variability present in speech features. The first term $p(\mathbf{x}_t|q_t)$ deals with the feature variability and describes the distribution of the speech features $\mathbf{x}_t$ emitted by the state q_t. The second term $p(q_t|q_{t-1})$ handles the variation on the speech rate by describing the temporal evolution of the states.

The parameters of this model are then the transition probabilities $a_{ij} = P(q_t = j|q_{t-1} = i)$ and the states distributions $b_j(\mathbf{x}_t) = p(\mathbf{x}_t|q_t = j)$. It can be noted that these parameters do not depend on the time frame, i.e., they are time-independent. The structure of a HMM is shown in Figure 3.2.

In the next two sections, we describe how to handle the two major problems that appear when using HMMs for ASR:

[1]This structure is also known as Bakis topology (Bakis, 1976).

- Given a training dataset, how to estimate the parameters that characterize the state distributions $b_j(\mathbf{x}_t)$ and the transition probabilities a_{ij} so that they can describe the temporal and feature variability present in the training data. This problem is known as training and it is discussed in Section 3.2.2.

- Given a sequence of observation vectors X and the sequence of states Q^W representing a linguistic unit W, how to compute the likelihood $p(X|W)$ expressed in (3.5). This problem is known as likelihood estimation and is explained in Section 3.2.3. In this thesis, the logarithm of this likelihood is referred to as acoustic score $J^W(X)$.

3.2.2 Training

As mentioned previously, the goal of training a HMM is to find the parameters $\Theta_a = (a_{ij}, b_j(\mathbf{x}_t))$ that better describe the variability contained in the speech features of a given training dataset. These parameters are estimated by optimizing an objective function on the training dataset. The most common criterion and also the one used in this thesis is maximum likelihood (ML) (Bahl *et al.*, 1983). There exist other criteria such as maximum a posteriori (MAP)[2] (Gauvin and Lee, 1992) or maximum mutual information (MMI) (Bahl *et al.*, 1986). When using the ML criterion, the parameters Θ_a are estimated as

$$\Theta_a = \arg\max_{\Theta} p(X|W, \Theta) \tag{3.6}$$

The major difficulty for estimating the HMM parameters is that the state associated to each feature vector (i.e., the acoustic region the feature vector belongs to) is not directly observable but must be inferred from the acoustic observations X. If we could know to which state $q_t = i$ is associated each speech feature $\mathbf{x}_t$, parameters defining the probability distribution of that state i could be directly estimated. Given a set of B training utterances containing $T_1, \cdots, T_b, \cdots, T_B$ frames respectively, transition probabilities would be simply estimated by counting the co-occurrences between states.

$$a_{ij} = \frac{\sum_{b=1}^{B} \sum_{t=1}^{T_b - 1} I_b(q_{t+1} = j, q_t = i)}{\sum_{b=1}^{B} \sum_{t=1}^{T_b} I_b(q_t = i)} \tag{3.7}$$

[2]This criterion is mainly used to adapt the HMM parameters to a particular speaker.

where $I_b(A)$ is the identity function that outputs one if event A is true in the utterance b and output zero otherwise. The parameters of the state emission distributions can be estimated by grouping all the feature frames belonging to the same region (i.e., state) and using the maximum likelihood criterion as follows

$$\Theta_{aj} = \arg\max_{\Theta} \prod_{\mathbf{x} \in \mathcal{X}(j)} p(\mathbf{x}|\Theta) \tag{3.8}$$

where $\mathcal{X}(j)$ denotes the set of speech features from the training data belonging to the state j. In this case, we are using the assumption that feature vectors are independent given the emission state.

Expectation-Maximization Algorithm

The procedure used to estimate the parameters of a HMM is based on the expectation-maximization (EM) algorithm (Dempster *et al.*, 1977; Moon, 1996). This algorithm increases the likelihood of a stochastic model that uses auxiliary variables. It consists of two steps, expectation (E) and maximization (M), that are repeated iteratively until convergence of the parameters:

1. E-step: The probability of the sequence of features X being in the region characterized by the state i at time frame t is estimated based on the current model parameters. This probability is referred to as state occupancy probability and is defined as

$$\gamma(t, i) = P(q_t = i|X, W) \tag{3.9}$$

2. M-step: The HMM parameters are estimated based on the state occupancies of the training samples computed in the previous step. The criterion for estimating the parameters maximizes the likelihood of each state distribution.

The EM algorithm applied to the estimation of the HMM parameters can be interpreted as a procedure where the acoustic regions specified by the states are iteratively redefined until convergence. Each region is characterized according to the features frames belonging to the that region.

Given a sequence of speech features X, the state occupancy $\gamma(t, i)$ estimated in the E-step indicates where the boundaries between regions appear along the sequence of features. This boundaries

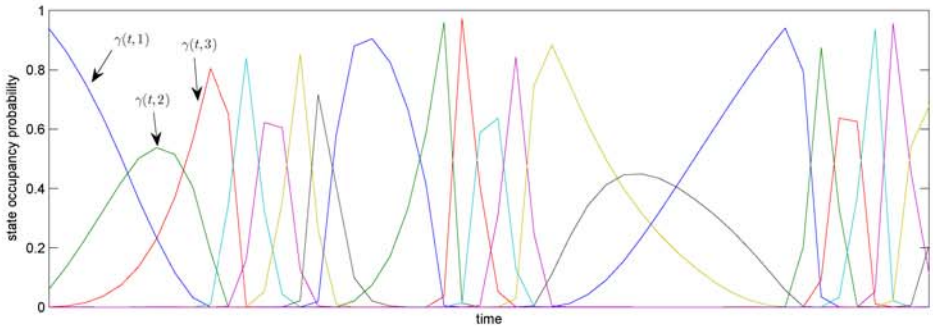

Figure 3.3. State occupancy probabilities along time. It can be noted that the sum of all occupancy probabilities at a given time frame t must be equal to one, i.e., $\sum_i \gamma(t,i) = 1, \forall t$.

can be either soft or hard. In the former case, the state occupancy take values than ranges between zero and one (i.e., boundaries are probabilistic) whereas hard boundaries refer to state occupancies that only take binary values: one or zero (i.e, boundaries are deterministic). The training algorithm that uses soft state occupancies is known as Baum-Welch. Figure 3.3 illustrates the evolution of state occupancy probabilities along time within a training utterance. When the boundaries between regions are hard, the state occupancy take values of one or zero. The training procedure that uses these sharp boundaries is then referred to as Viterbi training. In the following, we discuss both types of training methods.

Baum-Welch Training

The Baum-Welch (or forward-backward) algorithm uses two additional probabilities to estimate the state occupancy. At each time frame t, the alpha (or forward) $\alpha(t,i)$ and beta (or backward) $\beta(t,i)$ probabilities are computed. The advantage of these probabilities is that they can be obtained recursively. The alpha probability estimates the likelihood of the speech sequence until frame t, $X_{1:t}$, when the speech frame at that time, $\mathbf{x}_t$ is emitted by the state i given the linguistic message W. The alpha probability is thus defined as $\alpha(t,i) = p(q_t = i, X_{1:t}|W)$ and can be obtained recursively by

$$\alpha(t,i) = \sum_j \alpha(t-1,j) a_{ji} b_i(\mathbf{x}_t) \tag{3.10}$$

On the other hand, the beta probability estimates the likelihood of the feature sequence from the time frame $t+1$, i.e. $X_{t+1:T}$, given that the feature frame $\mathbf{x}_t$ is emitted by the state i and the linguistic message W. It is thus defined as $\beta(t,i) = p(X_{t+1:T}|q_t = i, W)$ and can be recursively obtained by

$$\beta(t,i) = \sum_j \beta(t+1,j) a_{ij} b_j(\mathbf{x}_{t+1}) \tag{3.11}$$

The state occupancy probability $\gamma(t,i)$ can then be estimated from the alpha and beta recursions in the following way

$$\gamma(t,i) = P(q_t = i|X,W) = \frac{p(X, q_t = i|W)}{p(X|W)} = \frac{\alpha(t,i)\beta(t,i)}{\sum_j \alpha(t,j)\beta(t,j)} \tag{3.12}$$

This probability can then be used in the M-step to estimate the parameters that maximize the likelihood.

Viterbi Training

Unlike the Baum-Welch training where the state occupancy is described by a probability distribution, in the Viterbi training the state occupancy is deterministic. Thus, $\gamma(t,i)$ can only be one or zero, depending if the frame $\mathbf{x}_t$ belongs to the region specified by the state i or not. The E and M steps in this approach are implemented as

1. E-step: Determine the most likely state sequence for each feature sequence X_b. This can be efficiently done by defining a modified version of the alpha recursion, where basically the sum is substituted by the maximum operator. Hence, only the most likely path is taken into account:

$$\alpha'(t,i) = \max_{Q_{1:t-1}} p(X_{1:t}, Q_{1:t-1}, q_t = i|W) \tag{3.13}$$

$$= \max_j \left[\alpha'(t-1,j) a_{ji}\right] b_i(\mathbf{x}_t) \tag{3.14}$$

   Once this recursion reaches the end of the utterance $\alpha'(T,i)$, the most likely sequence can be obtained by doing backtracking. This recursion uses dynamic programming and is called

Viterbi alignment (Viterbi, 1967). This alignment maps all the frames of the training dataset with the HMM states.

2. M-step: The parameters describing each state i are estimated from all the frames that have been assigned to that state in the E-step using the expression (3.8).

This process guarantees that the global likelihood over the training data increases after each iteration. Since only the most likely path must be consider, the Viterbi-based procedure is faster and simpler to implement than Baum-Welch training. It can be shown that both approaches yield similar results when the number of frames is sufficiently large (Merhav *et al.*, 1991).

3.2.3 Likelihood Estimation

Once the parameters defining the state distributions b_j and the transition probabilities a_{ij} are estimated, a likelihood $p(X|W)$ of a sequence of speech features X generated by the HMM representing a linguistic message W can be computed as (3.5). This can efficiently estimated by using the forward recursion defined in (3.10).

$$p(X|W) = \sum_{i=1}^{M} p(q_T = i, X|W) = \sum_{i=1}^{M} \alpha(T, i) \tag{3.15}$$

where T is the length of the sequence X and M is the total number of states.

In practice, the Viterbi approximation is applied to (3.15). This approximation assumes that the sum over $\mathcal{Q}^W$ can be approximated by the most likely path, i.e.,

$$p(X|W) \approx \max_{Q \in \mathcal{Q}^W} \prod_{t=1}^{T} b_{q_t}(\mathbf{x}_t) a_{q_{t-1} q_t} \tag{3.16}$$

Since the likelihoods $p(\mathbf{x}_t|q_t)$ are usually very small values, the logarithm of (3.16) is usually taken to avoid numerical problems.

$$\log p(X|W) \approx \max_{Q \in \mathcal{Q}^W} \left[\sum_{t=1}^{T} \log b_{q_t}(\mathbf{x}_t) + \log a_{q_{t-1} q_t} \right] \tag{3.17}$$

This expression can be efficiently evaluated by using the modified forward recursion presented in

3.14 as follows

$$\log p(X|W) \approx \max_{i \in (1,\cdots,M)} \log \alpha'(T,i) \tag{3.18}$$

where

$$\log \alpha'(t,i) = \max_j [\log \alpha'(t-1,j) + \log a_{ji}] + \log b_i(\mathbf{x}_t) \tag{3.19}$$

As we have mentioned before, the state distributions characterize the different acoustic regions that typically correspond to sub-word units such as phonemes. These regions are defined by irregular boundaries on the feature space. Hence, the probability distribution associated to the state i, $b_i(\mathbf{x}_t)$, must be able to characterize non-linear regions. In the next sections, we describe the most common models to estimate these probabilities.

3.2.4 HMM/GMM

The most common approach to model the state emission probabilities is a Gaussian mixture model (GMM). Each state i is thus characterized by a weighted sum of M normal distributions where each component m is parameterized with a weighting factor c_{jm}, mean μ_{jm} and covariance matrix Σ_{jm}.

$$b_i(\mathbf{x}_t) = \sum_{m=1}^{M} c_{im} \mathcal{N}(\mathbf{x}_t; \mu_{im}, \Sigma_{im}) \tag{3.20}$$

$$= \sum_{m=1}^{M} c_{im} \frac{1}{\sqrt{2\pi|\Sigma_{im}|}} e^{-\frac{1}{2}(\mathbf{x}_t - \mu_{im})^T \Sigma_{im}^{-1} (\mathbf{x}_t - \mu_{im})} \tag{3.21}$$

The main advantage of this model is that, given enough mixture components, GMM can estimate any probability distribution. Moreover, algorithms for estimating the parameters are well known.

The parameters of GMM can be estimated from the training data using again the EM algorithm. In this case, the auxiliary variable denotes the mixture component associated to each training feature frame. Hence, during the M-step for estimating the parameters of the HMM, another EM procedure is applied to estimate the parameters of the GMMs using the state occupancy $\gamma(t,i)$ expressed in (3.12). The two steps for the GMM training are

1. The E-step estimates $\gamma(t,i,m)$, the posterior probability for the feature vector $\mathbf{x}_t$ of belonging

to the Gaussian component m of state i.

$$\gamma(t,i,m) = \gamma(t,i)\left[\frac{c_{im}\mathcal{N}(\mathbf{x}_t;\mu_{im},\Sigma_{im})}{\sum_{m'=1}^{M} c_{im'}\mathcal{N}(\mathbf{x}_t;\mu_{im'},\Sigma_{im'})}\right] \tag{3.22}$$

2. During the M-step the parameters of the GMM are re-estimated using $\gamma(t,j,m)$ in the following way:

$$\hat{c}_{im} = \frac{\sum_{t=1}^{T}\gamma(t,i,m)}{\sum_{t=1}^{T}\sum_{m'=1}^{M}\gamma(t,i,m')} \tag{3.23}$$

$$\hat{\mu}_{im} = \frac{\sum_{t=1}^{T}\gamma(t,i,m)\mathbf{x}_t}{\sum_{t=1}^{T}\gamma(t,i,m)} \tag{3.24}$$

$$\hat{\Sigma}_{im} = \frac{\sum_{t=1}^{T}\gamma(t,i,m)(\mathbf{x}_t-\hat{\mu}_{im})(\mathbf{x}_t-\hat{\mu}_{im})^T}{\sum_{t=1}^{T}\gamma(t,i,m)} \tag{3.25}$$

Since the total number of parameters contained in a HMM/GMM-based system can be very large and the EM algorithm only guarantees convergence to a local maximum of the model likelihood. The training criterion must be carefully chosen to obtain reliable parameter values. There are two main strategies.

1. The number of mixture components is gradually increased. At each increment step, the initial parameters for the EM training algorithm are chosen based on the parameters estimated on the previous stage.

2. The training data is segmented using a clustering algorithm such as K-means. The data associated to each cluster is then used to estimate the initial parameters of the Gaussian distributions.

As mentioned before, a GMM can estimate any probability distribution given enough number of mixture components. However, the larger the number of Gaussian distributions, the higher the number of parameters to estimate and hence, more training data is required to obtain accurate parameters. In practice, the number of Gaussian distributions is limited by the amount of training data. Hence, speech features are transformed to reduce the complexity of the GMM. For example, speech features are usually decorrelated so that a diagonal covariance matrix can be applied. Also,

since GMM is a generative model, the parameter estimation does not penalize the rest of classes. Hence, it does not guarantee that wrong hypotheses yield higher likelihoods. In the next section, we present a different type of distribution estimator based on a discriminative classifier that can alleviate these limitations of the GMM.

3.2.5 Hybrid HMM/MLP

In Section 2.2.2, we describe the properties of MLP as a discriminative classifier and how it can estimate posterior probabilities of the classes represented by the outputs given the cepstrum-based speech features used as inputs. If we consider that the each HMM state is associated to a MLP output class, the state scaled emission probability of the state j can be estimated from the MLP-based posteriors and the class priors by using Bayes' rule in the following way

$$b_i(\mathbf{x}_t) = p(\mathbf{x}_t|q_t = i) = \frac{P(q_t = i|\mathbf{x}_t)p(\mathbf{x}_t)}{P(q_t = i)} \propto \frac{P(q_t = i|\mathbf{x}_t)}{P(q_t = i)} \tag{3.26}$$

the term $P(q_t = j)$ is the prior probability of the class j. The term $p(\mathbf{x}_t)$ can be ignored because it is constant for all the classes and .

The main advantages of using a MLP to estimate the state emission distributions are:

- The MLP is trained to be discriminative among the output classes. Hence, the scaled likelihoods are estimated to be maximum for the right class.
- The capability of the hidden layer to model high order moments helps in modeling correlation within an acoustic feature vector and across acoustic feature vectors over time when feeding the MLP with a context of speech features.
- As MLP is a non-linear classifier, it can better model the irregular class boundaries on the acoustic space.

The parameters to be trained are the weights of the MLP and the prior probabilities of the K output classes $(P(q_t = 1), \cdots, P(q_t = K))$. To train the MLP, we require the target output of the MLP. As in the case of the training the HMM, it can be deterministic or probabilistic. In the latter case, the target vector T_t at time frame t is the state occupancy probability $T_t = (\gamma(t, 1), \cdots, \gamma(t, K))$ defined in (3.12) (Hennebert *et al.*, 1997). In the deterministic case, the target vector is composed

by a one and zeros. The non-null component corresponds to the class at that time. The phoneme at each time can be obtained by doing Viterbi segmentation (Renals *et al.*, 1994). In both cases, the process is iterated until convergence. The prior probabilities can be estimated either by counting the number of times each phoneme is represented or by summing over the state occupancies.

The acoustic model score of a hybrid HMM/MLP model corresponding to the linguistic message W can be expressed as:

$$J_H^W(X) = \log p(X|W, \Theta_H) \approx \max_{Q \in \mathcal{Q}^W} \left\{ \sum_{t=1}^{T} \log \frac{P(q_t|\mathbf{x}_t)}{P(q_t)} + \log a_{q_{t-1}q_t} \right\} \tag{3.27}$$

where Θ_H denotes the set of parameters of the hybrid HMM/MLP, i.e., the transition probabilities, the class priors and the MLP weights.

As we have mentioned previously, the scaled likelihood of each HMM state is obtained from the posterior estimate of the associated MLP output divided by the corresponding prior. This presents two main limitations when using a large number of states, e.g, when using context-dependent phonemes.

- The large number of MLP outputs requires a huge number of weights. The number of parameters can be reduced by using some constraints on the structure of the MLP (Bourlard *et al.*, 1992). These constraints allow more robust estimates of the MLP parameters but decrease the modeling capability of the MLP.

- Since most of the classes would correspond to similar sounds, i.e., same phonemes in a different context, experiments have shown that a pure discriminative estimation of the posterior probabilities of context-dependent classes decreases the recognition accuracy (Cohen *et al.*, 1993).

Given the above limitations, hybrid HMM/MLP is typically used when modeling context-independent phonemes. We will see that this limitation can be overcome with the approach proposed in this thesis.

3.2.6 Discrete HMM

In the previous sections, the state distributions deal with continuous vectors on the real space. In this section, we present a form of HMM where state distributions are discrete. In this case, the continuous speech features are mapped on to a set of discrete values through a vector quantizer (VQ). Hence, for each sequence of speech features $X = (\mathbf{x}_1, \cdots, \mathbf{x}_T)$, a sequence of the same length T of discrete values corresponding to the cluster indexes $V = (v_1, \cdots, v_T)$ is obtained through the VQ. Each discrete value $v_t \in (1, \cdots, F)$ where F is the number of total clusters. The emission distribution for state i is a vector of F probabilities such that $\sum_{n=1}^{F} b_i(n) = 1$.

The main advantage of this type of HMM is that is very simple to implement and time decoding can be very fast since the state distribution can be simply obtained from a look-up table.

Vector Quantization

There are several methods to quantize the speech features, i.e., to find the optimal clustering of the speech features. We discuss two approaches: centroid-based methods and MLPs. The former approach can partition the feature space in a arbitrarily large number of clusters. The most known algorithm is known as K-means (Linde *et al.*, 1980). This technique selects the centroids based on a criterion that minimizes the global distance between the whole set of samples and the set of centroids. There is a trade-off regarding the number of centroids because the larger the number of clusters, the less the error of quantization but, on the other hand, the clustering is less robust because less training samples are associated to each cluster. In fact, the maximum number of clusters is determined by the total amount of training samples. Discriminative clustering approaches have also been investigated. In (Iwamida *et al.*, 1991), speech features are coded according to a Learning Vector Quantization (LVQ) (Kohonen, 1988). Unlike K-means, LVQ selects the centroids to maximize the discrimination between classes instead of minimizing the distortion error.

Clustering can also be done through a MLP (Lippmann and Gold, 1987). In this case, each MLP output corresponds to one cluster. Given a speech feature used as input, the cluster is defined as the MLP output with the highest value, i.e., the most probable class. Although the number of clusters cannot be as large as using the K-means approach and computational time is significantly higher, MLP can take advantage of its ability to model non-linear classification boundaries (Cerf

et al., 1994). Another advantage of MLPs is that they can provide fuzzy classification boundaries. This characteristic is exploited in the Chapter 6 where a generalized version of discrete HMM is presented.

Training and Decoding

Each state is characterized by a discrete probability distribution where each component corresponds to a cluster. These discrete distributions can be estimated by counting the co-occurrences between the states and the clusters and then normalizing so that the sum of probabilities is one. This can be iteratively done by using Viterbi training.

At the decoding stage, given a sequence of input labels $V = \{v_1, \cdots, v_t, \cdots, v_T\}$, the acoustic score is defined as

$$J_D^W(V) = \max_{Q \in \mathcal{Q}^W} \left\{ \sum_{t=1}^{T} \log P(v_t|q_t) + \log a_{q_{t-1}q_t} \right\} \tag{3.28}$$

where the emission probability of state q_t is represented as $P(v_t|q_t) = b_{q_t}(v_t)$.

It may seem that the use of discrete probability distributions does not make any assumption about the data distribution. While this is true, a strong assumption has however been made at the time of clustering, when defining the VQ model for modeling the clusters. In practice, discrete HMMs do not obtain as good accuracy as continuous HMMs. However, they are still used because of their low computational requirements and fast decoding since the emission probability can be simply obtained using a look-up table.

Unlike hybrid HMM/MLP, discrete HMMs can model context-dependent phonemes in a straightforward way. However, the main limitation of discrete HMMs is that relevant information for ASR can be lost during the quantization. In this thesis, we will present an approach that overcomes this limitation.

3.3 Template Matching

3.3.1 General Description

Template matching (TM) is a general classification technique that relies on the principle that a class W can be characterized by a set of samples (templates) $\mathcal{Y}(W)$ belonging to that class. For decoding a test sample X, a similarity measure $\varphi(X, Y)$ is computed between X and each template Y. The test sample X is then decided to belong to the same class as the template with the lowest similarity measure.

$$\hat{W} = \arg \min_{W \in \mathcal{W}} \min_{Y \in \mathcal{Y}(W)} \varphi(X, Y) \tag{3.29}$$

where $\mathcal{W}$ denotes the set of all possible linguistic messages. A critical point in TM is the choice of the distance measure between samples $\varphi(X, Y)$. This measure must be able to capture the properties of the feature parameters that characterize each class.

Since this technique is non-parametric, it does not make any explicit assumption about the data distribution. Hence, it can potentially yield better performance than parametric approaches, like HMMs. However, in general, it requires a large number of templates to properly characterize each class. This presents a limitation in terms of computational resources.

3.3.2 Dynamic Time Warping

In the case of ASR, the test sample is the sequence of speech features X and templates are sequences of speech features corresponding to linguistic messages. Traditionally, these features are based on the short-time spectrum but other speech representations such as posterior-based features can also be used. The score from the acoustic model $J_{TM}^{W}(X)$ can be computed as the minimum distance between the sequence of speech features X and all the templates corresponding to the linguistic message W.

$$J_{TM}^{W}(X) = - \min_{Y \in \mathcal{Y}(W)} \varphi(X, Y) \tag{3.30}$$

The similarity measure between samples $\varphi(X, Y)$ used in ASR is known as dynamic time warp-

ing (DTW) and minimizes the global distortion between two sequences. Given a test sequence $X = \{\mathbf{x}_1, \cdots, \mathbf{x}_T\}$ and a template sequence $Y = \{y_1, \cdots, y_{T_Y}\}$, the DTW-based distance is defined as

$$\varphi(X, Y) = \min_{\phi} \sum_{t=1}^{T} d(\mathbf{x}_t, y_{\phi(t)}) \tag{3.31}$$

where $d(\mathbf{a}, \mathbf{b})$ represents the local distance between two vectors $\mathbf{a}$ and $\mathbf{b}$. This function is typically Euclidean or Mahalanobis distance but, depending on the choice of the speech features, other more suitable similarity measures can be used. For instance, if posterior features are used to form the templates and the test utterances, information theoretic measures can better describe the local similarity between features.

The function ϕ maps the vectors from the templates to the vectors in the test utterance. The following constraints are typically used in the mapping function

$$\begin{aligned} \phi(1) &= 1 \\ \phi(T_Y) &= T \\ 0 \leq \phi(t) - \phi(t-1) &\leq 2 \end{aligned} \tag{3.32}$$

where T_Y is the number of frames in the template Y. These constraints guarantee that no more than one template frame will be skipped for each test frame and also, that every test frame is matched only with one template frame. DTW is similar to the Viterbi approximation of the likelihood estimation of HMMs. It can be observed that expressions (3.31) and (3.17) are equivalent if state transition probabilities are considered uniform. Hence, a similar recursive procedure based on dynamic programing as described by expressions (3.18) and (3.19) can be applied to compute (3.31).

3.3.3 Comparison with HMM

DTW used in TM and Viterbi algorithm used for HMM decoding are two instances of dynamic programming in which a global score is found as a sum of local distances along the optimal alignment between the input and reference. By definition, the optimal alignment is the one that yields the smallest distortion (or best match) between the input and the reference. The main conceptual difference is that in the case of DTW the reference is a data stream like the input, while in the case of

Viterbi, the reference is a HMM. These differences primarily have an impact on the way the local distances are computed.

In this section, we analyze the similarities and differences between TM and HMM. We start from the Viterbi formulation of HMMs and show how the HMM states can be interpreted as frames of a template and vice versa. We first consider that the emission distribution of each state i is modeled by a single Gaussian $\mathcal{N}(\mathbf{x}; \mu_i, \Sigma_i)$. Moreover, all transition probabilities are pooled to a single set with only three different probabilities: self-loops, successor transitions and state skips. Under these constraints, we can write the Viterbi expression in (3.17) as

$$
\begin{aligned}
-\log p(X|W) &\approx \min_{Q \in \mathcal{Q}^W} \left[\sum_{t=1}^{T} \log \mathcal{N}(\mathbf{x}_t; \mu_{q_t}, \Sigma_{q_t}) - \log a_{q_{t-1} q_t} \right] && (3.33) \\
&= \min_{Q \in \mathcal{Q}^W} \left[\sum_{t=1}^{T} (\mathbf{x}_t - \mu_{q_t})^T \Sigma_{q_t}^{-1} (\mathbf{x}_t - \mu_{q_t}) + \log 2\pi |\Sigma_{q_t}| - 2 \log a_{q_{t-1} q_t} \right] && (3.34) \\
&= \min_{Q \in \mathcal{Q}^W} \sum_{t=1}^{T} (\mathbf{x}_t - \mu_{q_t})^T (\mathbf{x}_t - \mu_{q_t}) && (3.35) \\
&= \min_{Q \in \mathcal{Q}^W} \sum_{t=1}^{T} d(\mathbf{x}_t, \mu_{q_t}) && (3.36)
\end{aligned}
$$

In (3.35), we consider that the covariance matrix is fixed for all the states. Then, the Mahalanobis distance can be turned into the Euclidean distance by applying a linear transformation to the input feature vector. Also, transition probabilities for self-loops, successor transitions and state skips are assumed uniform. When comparing (3.36) with (3.31), we can observe that both expressions are equivalent when the mean states μ_{q_t} are the features composing the template y_i and the mapping $\phi(t)$ is represented as the state path $\{q_t\}$.

Hence, the similarity between HMM and DTW can be described as

- The equivalence is valid for continuous-density HMMs using single normal distributions.
- The frames of the reference template Y correspond to the means of states in the HMM.
- The negative log-probability in HMMs corresponds to local distances in the DTW framework.
- Using Euclidean distance is equivalent to using HMM states with pooled unity covariance matrix.

- Transition probabilities as well as the topology of HMM states have a similar role as DTW path constraints.

However, the main differences between these two approaches are:

- Typically, the number of states in an HMM is considerably smaller (factor 2-3) than the number of frames in the input, while in TM the input X and reference Y are roughly of equal length. In fact, the skipping state transition is allowed in TM to handle the situation where the template is longer than the test sequence.

- In HMMs, duration is modeled through the state transition probabilities yielding to an exponential distribution. This assumption is not very accurate and alternative distributions have been investigated. In (Bourlard and Wellekens, 1986), duration is explicitly modeled through a Poisson distribution. When applying templates, these transition probabilities are typically uniform but duration of a linguistic unit corresponds to the length of the sequence of vectors. Thus, duration is better described when using TM.

- HMM states represent a particular linguistic unit (e.g. phonemes or context-dependent phonemes) while template frames are not associated to any acoustic label. Hence, HMMs can better describe the region of the feature vectors associated to the linguistic unit.

- Since the parameters of the HMM are estimated by assuming that states are independent of each other (see expression (3.8)), the temporal continuity of speech features is better modeled by TM than HMM because templates are sequence of frames corresponding to actual speech utterances.

- Since the emission distributions of each state are independent, there is no way to impose continuity between the Gaussian distributions from different states. Thus, an observation sequence can be recognized using a sequence of mixtures which has never been observed in the training set (Illina and Gong, 1998). This phenomenon does not happen within TM framework because states within a template are sequences from the training data.

- The use of characteristics of the speech signal that are not directly related to the linguistic message can be easier to apply in templates than in HMMs. These meta-linguistic features

mainly refer to the speaker and the environment. The use of these characteristics can potentially improve the accuracy and reduce the complexity of the ASR systems. For example, in (Aradilla *et al.*, 2006a) the pitch frequency is used as meta-linguistic information for clustering the templates. Thus, the meta-linguistic information can be considered as prior knowledge at the decoding stage. Experiments show that better ASR performance can be obtained using this approach. Moreover, the decoding time is significantly reduced since the number of templates evaluated is fewer.

3.3.4 Current TM-based Trend (Episodic Modeling)

As the size of speech databases has been increasing and computational resources have become more powerful, the complexity of HMM-based approaches has also been scaled proportionally but their ASR accuracy has not improved at the same level. As an alternative approach to HMMs, TM has recently gained new attention by the ASR community (Wachter *et al.*, 2003; Axelrod and Maison, 2004; Aradilla *et al.*, 2005; Wachter *et al.*, 2007). The main idea is to use real data instead of increasing the capacity of parametric models. This is motivated by the results on two different fields:

- Experiments on the field of human perception suggest that patterns are not described as abstract entities by the brain. Instead, patterns are associated to individual samples (episodes) containing all the information of the situation (Goldinger, 1998). When an element is presented, the most similar episodes are activated. This element is then classified depending on the information provided by the selected episodes. We can compare the paradigm of the abstract entity with a parametric model representing a class and the episodic theory with a TM approach where episodes are templates (Strik, 2003).

- In speech synthesis, parametric models that synthesize the speech waveform are outperformed by the technique of using real intervals of waveforms from the training data (Hunt and Black, 1996). Thus, the main issue of this technique is not the choice of a suitable model for the speech signal but finding an appropriate cost function between intervals so that the global waveform is satisfactory perceived (Wouters and Macon, 1998; Hunt and Black, 1996; Vepa *et al.*, 2002).

The crucial issue of this method is to efficiently search in the huge space of all the possible templates. Bottom-up search methods (Wachter *et al.*, 2003) and the use of meta-linguistic information (Aradilla *et al.*, 2006a) are strategies that have been successfully applied.

3.4 Summary

In this chapter, we have described the two main approaches, templates and HMMs, for modeling the parameters features extracted from the speech signal and computing the score corresponding to the acoustic modeling, $J^W(X)$. Both methods are similar because they consist of reference sequences that must be adapted to the test sequence X. In fact, a template can be seen a particular case of a HMM where each state corresponds to a frame forming the template.

Posterior features can be used in all the acoustic models discussed in this chapter:

- They can be used as features for HMM/GMM. This situation is explained in detail in the next chapter.
- In hybrid HMM/MLP, they are used as estimators of the state emission likelihood.
- In discrete HMMs, posteriors can be turned into labels by selecting the class with the highest probability.
- When using templates, posteriors can be used to form both the templates and the test utterances. Traditional measures, such as Euclidean distances, can be used to compute the similarity between vectors.

One of the main contributions of the present thesis consists in describing a general framework that unifies the above described posterior-based acoustic models.

Chapter 4

Posterior-based ASR Systems

The exploitation of posterior probabilities have recently gained a lot of attention in the ASR community given their optimal classification properties. They have been used in a large number of applications ranging from confidence measures (William and Renals, 1999) and beam search pruning (Abdou and Scordilis, 2004) to word lattice re-scoring (Mangu *et al.*, 2000). This chapter presents a survey of ASR systems using posterior probabilities of sub-word units. We focus on those systems that incorporate the use of posteriors in either the feature extraction process or the structure of the acoustic model.

The criterion used in this chapter for classifying the different approaches is based on the method for estimating the posterior probabilities. Each section is focused on a particular estimation model that use a different information for estimating the posteriors.

4.1 Multi-Layer Perceptron

The multi-layer perceptron (MLP) is a special type of artificial neural network that constitutes the most common method for estimating sub-word posterior probabilities. The reasons are that it scales well with large amount of training data and can possibly introduce a long temporal context. It has been explained in Section 2.2.2 and is the method that will be used in this work for estimating the sub-word posterior probabilities. Other types of neural networks such as recurrent neural networks (RNN) can also be used to estimate posteriors (Santini and Braun, 1995; Hochberg *et al.*,

1995; Robinson, 1994).

Given the good properties of MLP for estimating the sub-word posterior probabilities, the role of MLP-based posteriors involve three different cases: features, scores and labels:

Features : A vector is formed from the estimates of posterior probabilities over the set of classes. This vector is then used as input to state-of-the-art acoustic models (mainly HMM/GMM-based systems). The posterior vector is usually post-processed so that it can better fit the modeling assumptions of the GMM. This is the dominant role of posterior features because it benefits from the advantages in terms of scalability and efficiency of HMM/GMM-based approaches.

Scores : Posterior probabilities are used to compute the emission likelihood of HMM states. In this case, the posterior classes must correspond to the same representation of the HMM states, which are typically phoneme-level units. This type of hybrid HMM replaces the traditional approach for computing the state emission likelihoods (GMMs) with a model that provides more discriminant classification.

Labels : This role is used in discrete HMMs. A model for estimating posterior probabilities is used to quantize standard spectral-based speech features. The codeword (label) of each feature vector is defined as the most probable posterior class. This approach is usually chosen for its simplicity and easy implementation since the computation of the state likelihood simply consists on the access to a look-up table.

4.1.1 Features

Basic Configuration: Tandem Approach

Posterior features estimated from a MLP can be used as inputs to a HMM/GMM-based system. However, given the particular properties of posterior features, the state emission likelihood distributions of posterior features can be difficult to model through a mixture of Gaussian distributions. Hence, posterior features are processed to make them more suitable to be modeled by a GMM. This post-processing involves

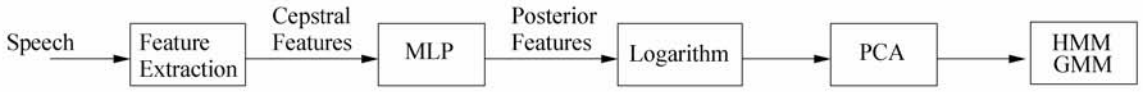

Figure 4.1. Standard approach for deriving and using tandem features. The phoneme posterior vectors are estimated using a MLP. These posteriors are "Gaussianized" and decorrelated by taking the logarithm and PCA transformation. The result of this transformation is used as input features for state-of-the-art HMM/GMM-based systems.

1. A non-linear transformation (typically the logarithm) so that posterior values follow an probability distribution more similar to the normal distribution.

2. Feature components are decorrelated through a principal component analysis (PCA) transformation derived from a training dataset. Hence, Gaussian distributions using diagonal covariance matrix can be used. Since this transformation implies a weighted sum of the log-posteriors, the features obtained from the PCA output are even more similar to the normal distribution.

After performing the steps described above, the processed posterior features are more suitable to be modeled by a GMM. This approach is known as tandem (Hermansky *et al.*, 2000a) and the scheme of this system is illustrated in Figure 4.1.

The advantage of this system is that it exploits the discriminative properties of the MLP-based posterior features and the good modeling capabilities of GMMs. However, the meaning of the components of the posterior features is lost during the post-processing step. Also, GMM does not take into account the particular properties of posterior features. As a consequence, a large number of parameters for describing the Gaussian distributions of each state mixture is required. Usually, a dimensionality reduction is performed through the PCA to the post-processed posterior features and the reduced processed posterior feature is appended to standard spectral-based features to yield further improvement.

Exploiting the Temporal Context

One of the main advantages of using a MLP for estimating the posterior features is that it can easily incorporate a long temporal context. This can be simply done by feeding the MLP with a set of adjacent input features as shown in Figure 2.3. In the standard tandem approach, a context involving 9 frames (4 frames per side) is generally used. This corresponds to roughly 90ms.

A more complex MLP-based architecture has been investigated so that it can take a longer tem-

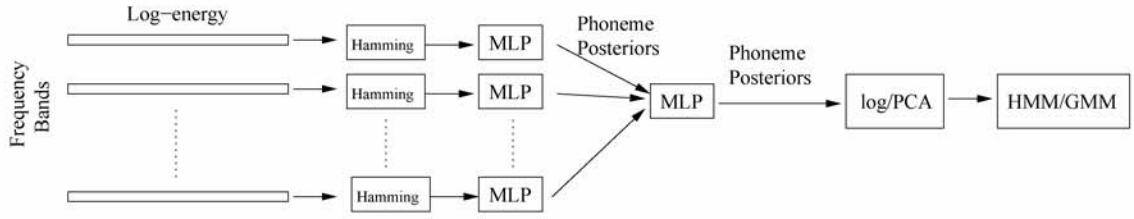

Figure 4.2. Scheme of the TRAPs system. The log-energy of 1-second temporal window is first computed for each frequency band. Typically, 15 bands are used. Since the shifting period of each energy computation is 10ms, a sequence of 100 log-energy values is weighted by the Hamming window and used as input to a MLP. The output of each MLP estimates phoneme posterior probabilities. The 15 posterior outputs are then turned into log-likelihoods and concatenated to form a global vector that is used as input for a merger MLP. The resulting posteriors are post-processed and used as inputs to standard HMM/GMM as in tandem approach.

poral context as input for estimating the posterior features (Hermansky *et al.*, 2000b). In this case, one second is taken into account. First, the log-energies of each frequency band are weighted by a Hamming window. Each sequence is then used as input to an independent MLP that estimates phoneme posterior probabilities. The output of each MLP is then combined by using another MLP. In this case, the input of the merger MLP is the concatenated vector of the log-likelihoods of the phonetic classes. The output of this merger MLP are again estimates of phoneme posterior probabilities. It can be noted that every MLP is characterized by a low capacity and hence, the number of total parameters is equivalent to standard tandem system. This system is known as "TRAPs" and its scheme is illustrated in Figure 4.2.

The motivations of this approach are two-fold:

- The use of multi-band processing: This technique computes early feature extraction techniques in individual spectral sub-bands independently. Such strategy then allows for selective attenuation of the results from unreliable sub-bands and is reported to be more robust in the presence of frequency-selective noise (Hermansky *et al.*, 1996).

- Incorporating a long temporal context: A longer temporal information can yield more reliable phoneme posterior estimates. In fact, an analysis of a large hand-labeled database has shown that the information about the underlying phonetic class is spread over a considerable interval in time (Yang *et al.*, 1999).

The above structure has been further developed in other works. For instance, in (Hermansky and Fousek, 2005), the "m-rasta" approach is presented. The derivative function of the Gaussian distribution is used to weight the log-energies. By choosing different variances of the derivative

filters, different temporal resolutions and modulation frequencies are selected. This information is then concatenated and used as input for a MLP that estimates phoneme posteriors. In the approach known as HATS (hidden activation TRAPs) (Chen *et al.*, 2004), the output of the hidden layers are concatenated and used as inputs for the merger MLP that estimates phoneme posteriors. The motivation of this technique lies on the fact that the phoneme posteriors of each frequency band can be very similar among them. This redundancy can be alleviated by using the projection of the input log-energies onto a higher dimensional space represented by the hidden layer of the MLP. This approach has further investigated in (Grezl *et al.*, 2007), where the MLP is used to reduce the dimensionality of the hidden layer while keeping relevant information for the merger classifier. In this case, a 5-layer MLP is considered where the middle layer acts as a bottle-neck function and is used as input for the MLP estimating phoneme probability estimates.

4.1.2 Scores

Hybrid HMM/MLP (Bourlard and Morgan, 1993) approaches were probably among the first ones to make extensive use of posterior probabilities in ASR. In this case, MLPs are used to estimate the emission probabilities required in HMM systems (see Section 3.2.5).

Global Training

The parameters of the MLP and HMM are generally trained independently, using different objective functions over the training data. A research direction has focused on unifying the training criteria of both MLP and HMM through a global optimization. In (Bengio, 1993), the likelihood of the combined system is maximized by applying a gradient-ascent technique. In this case, the forward-backward algorithm (see Section 3.2.2) implicitly computes the derivative of the likelihood with respect to the emission probabilities of the HMM. This derivative can then be applied for estimating the parameters of the MLP.

Context-dependent Classes

As described in Section 3.2.5, one of the major limitations of hybrid HMM/MLP is that HMM states must represent the same linguistic unit as the posterior classes. Hence, modeling context-dependent phonemes implies training a MLP with a huge number of outputs. This direct approach

would cause problems in terms of system complexity and reliable estimation of the MLP weights. In (Bourlard *et al.*, 1992), it is shown that the context-dependent posterior can be modeled with MLPs that are not substantially larger than context-independent MLPs. Let us define c_k, c_k^l, c_k^r and $\mathbf{x}_t$ as the central phoneme, the left and right phonetic contexts and the observation vector respectively. Then, the posterior probability of the context-dependent phoneme can be rewritten as

$$P(c_k^l, c_k, c_k^r|\mathbf{x}_t) = P(c_k^l|c_k, c_k^r, \mathbf{x}_t)P(c_k^r|c_k^l, \mathbf{x}_t)P(c_k|\mathbf{x}_t) \tag{4.1}$$

where the three independent posterior terms can be estimated through MLPs trained individually. It can be noted that the last term $P(c_k|\mathbf{x}_t)$ corresponds to the output of the standard MLP estimating phoneme posterior probabilities.

In (Cohen *et al.*, 1993), a different factoring strategy is employed to estimate the posterior probabilities of context-dependent phonemes. In this approach, the weights connecting the input and the hidden layer are shared among all the possible contexts. Initially, the MLP is trained using context-independent targets. During a second training stage, targets correspond to context-dependent phonemes and the weights from the hidden to the output layer are estimated. The outputs, then, correspond to context-dependent classes represented by the HMM states. This strategy both reduces the total number of MLP weights and smooths the estimates of the context-dependent phonemes by initially training the MLP using context-independent classes.

A different factoring product is used in (Leung *et al.*, 1992). In this case, the context-dependent conditional posterior probability is decomposed as

$$P(c_k^l, c_k, c_k^r|\mathbf{x}_t) = P(c_k|c_k^l, c_k^r, \mathbf{x}_t)P(c_k^l|c_k^r, \mathbf{x}_t)P(c_k^r|\mathbf{x}_t) \tag{4.2}$$

where the posterior terms are estimated by simple networks. This posterior estimate is used in the context of segmental modeling (Ostendorf *et al.*, 1996).

In (Fritsch and Finke, 1998), the posterior probability of the context-dependent classes is factorized using a hierarchical clustering process. The set of context-dependent phonemes $\{s_k\}$ is grouped into subsets $\{S_i\}$. Therefore, we can rewrite the posterior probability of the context-dependent class

s_k as a joint probability of state and appropriate subset S_i. Then, it can be factorized as

$$P(s_k|\mathbf{x}_t) = P(s_k, S_i|\mathbf{x}_t) \quad \text{with} \quad s_k \in S_i \tag{4.3}$$

$$= P(S_i|\mathbf{x}_t)P(s_k|S_i, \mathbf{x}_t) \tag{4.4}$$

Thus, the global task of discriminating between all the context-dependent phonemes has been converted into discriminating between subsets S_i and discriminating between states s_k contained within each of the subsets S_i. Each of these factors can then be estimated by a neural network.

The approaches described above are based on the factorization of the posterior probability of context-dependent classes so that each factor can be estimated from a simpler neural network. In (Rottland and Rigoll, 2000), context-dependent phonemes are modeled by using a different approach. The emission probability of each state is based on a mixture of distributions as expressed in (3.20).

$$b_i(\mathbf{x}_t) = \sum_{m=1}^{M} c_{im} p(\mathbf{x}_t|u_{im}) \tag{4.5}$$

where u_{im} denotes the mth component of the mixture corresponding to the state i. In this case, the set of emission distributions $\{p(\mathbf{x}_t|u_{im})\}$ is fixed for all the HMM states and the parameters that characterize each state are the weighting factors $\{c_{im}\}$. Hence,

$$b_i(\mathbf{x}_t) = \sum_{m=1}^{M} c_{im} p(\mathbf{x}_t|m) \tag{4.6}$$

The emission distributions $\{p(\mathbf{x}_t|m)$ can then be estimated from a posterior as in standard hybrid HMM/MLP systems. Thus,

$$b_i(\mathbf{x}_t) = \sum_{m=1}^{M} c_{im} \frac{p(m|\mathbf{x}_t)}{p(m)} \tag{4.7}$$

where the posterior classes $\{m\}$ represent phonemes or any other sub-word unit. The state parameters $\{c_{im}\}$ can then be estimated using a standard maximum likelihood criterion and the EM training algorithm.

Using Long Temporal Context

Most of the approaches described in the previous section for estimating posterior probabilities using a longer temporal context can be used for estimating the emission likelihood of the HMM states. In (Hermansky and Sharma, 1999), experiments are reported where TRAPs-based posteriors are used as scores for hybrid HMM/MLP. This type of posteriors is combined using a merger MLP with standard short-term posteriors. This combination results in a significant improvement in both clean and noisy conditions.

More recently, a different strategy for exploiting the temporal context has been investigated. In (Pinto *et al.*, 2008), a MLP is used in the standard fashion to estimate phoneme posterior probabilities. The output of this MLP is then grouped in a vector to form a posterior feature. Then, a sequence of posterior features is used as inputs to a merger MLP that estimates phoneme posterior probabilities. This merger MLP exploits the information contained on the temporal sequence of posterior features.

4.1.3 Labels

Artificial neural networks (ANNs) have been applied for generating codebooks for discrete HMMs. Early attempts rely on learning vector quantizer (LVQ) (Iwamida *et al.*, 1991) as an effective neural alternative to standard clustering algorithms. The main advantage of this approach with respect to traditional methods, such as K-means (Linde *et al.*, 1980), is that it follows a discriminative procedure to define the centroid of the clusters. The network topology of LVQ can be seen as MLP with no hidden layer. The output values $\{z_i\}$ are defined as

$$z_i = ||\mathbf{w}_i - \mathbf{x}||^2 \tag{4.8}$$

where $\{\mathbf{w}_i\}$ and $\mathbf{x}$ denote the weights and the input vector respectively. In this case, weights correspond to the set of codewords (codebook) of the system. The activation function expressed in (4.8) expresses a distance between the input vector $\mathbf{x}$ and the ith codeword of the codebook. The training criterion for estimating the weights $\{\mathbf{w}_i\}$ minimizes the classification error (McDermott and Katagiri, 1994). This approach can be further improved by increasing the capacity of the LVQ. This is performed by incorporating a hidden layer in the topology of the ANN (Rigoll, 1994) using the

maximum mutual information as a training criterion.

MLP can also be used as labelers (Ma and Compernolle, 1990). Each observation vector is used as input for a MLP trained to estimate posterior probabilities. The codeword (label) of each input vector is defined as the most probable posterior class. This discrete label is then used as input for the discrete HMM (see Section 3.2.6). This system can be successfully applied mainly for small vocabulary tasks, where a model for each word is used (instead of phoneme-level models). The basic model was extended in (Cerf *et al.*, 1994) with the introduction of *N-top*, that is to say, the N-best (most probable) MLP outputs are considered (not only one) and passed to the HMM. This is of particular interest in the regions of the feature space in proximity of the boundary between the posterior classes, where mis-classification is more likely.

4.2 Support Vector Machines

Support vector machines (SVMs) is a discriminant classification method that learns the linear decision boundary between two classes (Vapnik, 1995). The criterion for estimating this classification boundary is to maximize the distance between the two sample datasets describing the classes. Since in practice, classes are not linearly separable, features are transformed to a higher dimensional space.

One possible use of SVMs is the estimation of posteriors probabilities that replace the emission likelihood traditionally estimated by a GMM. In (Ganapathiraju *et al.*, 2000), the output of this classifier is turned into a posterior using the sigmoid function (Platt, 2000)

$$P(c_k|\mathbf{x}_t) = \frac{1}{1+\exp(Af_k(\mathbf{x}_t)+B)} \tag{4.9}$$

where $f_k(\mathbf{x}_t)$ is the distance between the observation vector $\mathbf{x}_t$ and the boundary decision describing the class c_k. The free parameters A and B are estimated from a cross-validation data to avoid severe bias on the training data.

The non-linear transformation for turning the features to a high-dimensional space can be chosen so that the output of the linear classifier provides a meaningful interpretation. In particular, if the kernel Fisher discriminant is applied, the outputs are estimates of the posterior probabilities

of the classes when assuming that classes follow a Gaussian distribution (Mika *et al.*, 1999). This convenient property is applied in (Salomon *et al.*, 2002) where phoneme posterior probabilities are estimated from a SVM using the kernel Fisher discriminant.

4.3 Maximum Entropy Models

The maximum entropy (MaxEnt) criterion (Jaynes, 1982; Berger *et al.*, 1996) can be used to estimate probability distributions. MaxEnt modeling is based on the principle of integrating priors as constraints. It states that the modeled probability distribution should be consistent with a set of constraints and otherwise be as uniform as possible. In other words, MaxEnt modeling attempts to make the best use of all the available constraints without making any assumptions about the conditions which are not present. Another advantage of this approach is that it can easily incorporate information from independent sources by just imposing new constrains. The posterior probability of the class c_k follows the formulation

$$P(c_k|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp \sum_i \lambda_i f_i(c_k, \mathbf{x}) \tag{4.10}$$

where $\{f_i(c_k, \mathbf{x})\}$ are the set of constraint functions satisfying the context x when the class c_k occurs. The MaxEnt formulation can be related to the criterion used by the MLP when using the softmax function to define the output value as described in (2.6) (Bourlard and Wellekens, 1990).

In (Hifny and Renals, 2005), phoneme posterior probabilities are used in a phoneme recognition system. The set of constraints for estimating each distribution is defined as the likelihood values obtained from a set of Gaussian distributions. These posteriors can then be used to estimate scaled likelihoods in the same way as hybrid HMM/MLP by dividing by the phoneme priors as expressed in (3.26).

Direct Models

This approach can be used in the so-called *direct models* (McCallum *et al.*, 2000). Unlike generative HMMs, that rely on the likelihood $P(X|W)$ of a sequence of speech features X being generated by a model W, direct models explicitly compute the posterior probability $P(W|X)$. Based on some

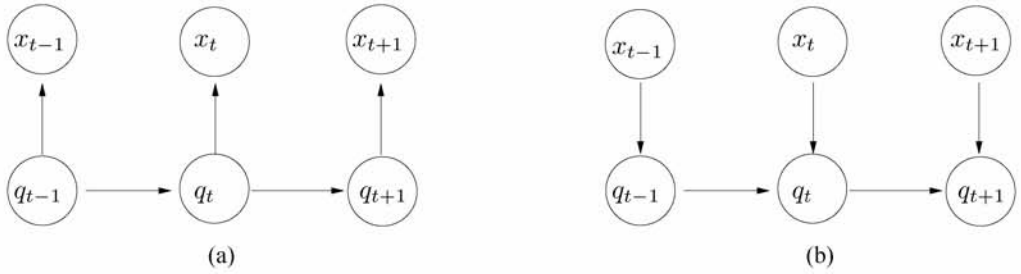

Figure 4.3. Figure (a) shows the structure of the graphical model followed by traditional HMMs. Figure (b) illustrates the structure used by direct models. It can be seen that, unlike HMMs, direct models are not a generative approach.

independence assumptions, this expression can be reformulated as

$$P(W|X) = P(q_1, \cdots, q_T|\mathbf{x}_1, \cdots, \mathbf{x}_T) = \prod_t P(q_t|q_1, \cdots, q_{t-1}, \mathbf{x}_t) \tag{4.11}$$

In theory, the term $P(q_t|q_1, \cdots, q_T, \mathbf{x}_t)$ can be computed using MaxEnt modeling by applying constrains satisfying the condition $q_t|q_1, \cdots, q_T, \mathbf{x}_t$. However, a simplification must be done to avoid the problem of unreliable estimates because of the sparse data. In (Likhododev and Gao, 2002; Kuo and Gao, 2006), the term $P(q_t|q_1, \cdots, q_{t-1}, \mathbf{x}_t)$ is simplified to $P(q_t|q_{t-1}, \mathbf{x}_t)$. Figure 4.3 compares the state structure of traditional HMMs and direct models.

The main limitation of this approaches is that MaxEnt modeling is difficult to manage if hidden variables occur. Hidden variables may lead to non-convex objective function and hence, training algorithms cannot be applied. However, a modified version of the training algorithm for MaxEnt modeling that can discriminatively estimate the parameters of the state Gaussian densities is developed in (Macherey and Ney, 2003).

Conditional Random Fields

Conditional random fields (CRFs) (Lafferty *et al.*, 2001) are generalizations of the MaxEnt models where the constraint functions represent a particular structure. This structure can be described as a graphical model where the probability of each state depends on its neighbourhood.

The above MaxEnt approaches attempt to model the posterior probability of a single state q_t. This is because the HMM is restrictive in that all the states need to model the observations in a uniform way, and that it is difficult to incorporate long-range dependencies between the states

and the observations. To remedy this limitation, CRFs can estimate the posterior probability of an entire state sequence $Q = \{q_1, \cdots, q_T\}$ given the observation sequence X using an exponential distribution similar to MaxEnt model (Gunawardana *et al.*, 2005)

$$P(W|X) = \frac{1}{Z(X)} \sum_{Q \in W} \exp\left[\lambda_Q f(W, Q, X)\right] \tag{4.12}$$

where the parameters $\{\lambda_Q\}$ of the constraints functions $f(W, Q, X)$ are learned from a training dataset.

Recently, CRFs have been used to estimate phoneme posterior estimates that have been applied in a HMM/GMM-based system in the same way as tandem (Fosler-Lussier and Morris, 2008). This approach takes advantage of both the good modeling capabilities of CRFs and the efficient training and decoding algorithms of HMM/GMM.

4.4 Gaussian Distributions

Posterior probabilities can also be estimated from generative models, such as Gaussian distributions, following Bayes' rule.

$$P(c_k|\mathbf{x}) = \frac{p(\mathbf{x}|c_k)P(c_k)}{p(\mathbf{x})} = \frac{p(\mathbf{x}|c_k)P(c_k)}{\sum_{c'} p(\mathbf{x}|c'_k)P(c'_k)} \tag{4.13}$$

where the term $p(\mathbf{x}|c_k)$ is the likelihood of the observation x given the class c_k and $P(c_k)$ represents the prior probability.

The main advantage of this estimation method when compared to the previous ones is that a large number of posterior class can be easily used by applying a non-supervised training criterion using the EM algorithm. It can be shown that high dimensional spaces are more likely to be linearly separable than low dimensional spaces (Cover, 1965). However the EM training algorithm does not follow a discriminative procedure and hence, can yield to a low classification performance. Posteriors estimated from a GMM and from a MLP are compared in (Reyes-Gomez and Ellis, 2002). This study shows that MLP-based posterior estimates significantly outperform GMM-based posteriors because of the discriminative training and the projection onto the high-dimensional space performed by the hidden layer of the MLP.

In (Povey *et al.*, 2005), a discriminative step is incorporated after the generation of GMM-based posterior probabilities. In this approach, a mixture of thousands of Gaussian distributions is estimated from the training dataset. A high dimensionality posterior vector is then formed using the expression (4.13) and assuming uniform class priors. A dimensionality reduction is then performed through a linear transformation characterized by a matrix M. The elements of this matrix are estimated by optimizing an objective function that minimizes the phoneme error rate. The output of this transformation is added to the original feature vector $\mathbf{x}_t$.

$$\mathbf{x}'_t = \mathbf{x}_t + M\mathbf{h}_t \tag{4.14}$$

where $\mathbf{h}_t$ denotes the posterior feature lying on the high dimensional space. The resulting vector $\mathbf{x}'_t$ is used as input to a HMM/GMM system. Recently, this approach has been generalized in a framework where features are not transformed but the parameters of the GMM characterizing each state (Sim and Gales, 2007). In this way, this approach can be seen as semi-parametric trajectory model where the parameters of the GMMs are updated to better describe the trajectory of the observation vectors.

4.5 k-Nearest Neighbours

The above methods are all parametric models. Hence, they make some assumptions about the estimation through the structure of the method. Posterior probabilities can also be estimated through non-parametric techniques. A non-parametric estimation makes no assumption of the targeted probability distribution shape (Duda *et al.*, 2001). In (Lefèvre, 2003), the k-nearest neighbours (k-NN) technique (Fukunaga, 1990) is applied to estimate the posterior probability of HMM states. The state posterior is then estimated as[1]

$$p(c_k|\mathbf{x}) = \frac{k_c}{k} \tag{4.15}$$

where k_c are the number of samples belonging to the class c among the k closest training samples to the observation vector $\mathbf{x}$. The average classification error of this technique is always less than

[1]It must be noted that sub-index of c_k is not related to the parameter k of the non-parametric approach.

twice the optimal Bayesian error. In addition, the error probability decreases monotonically when increasing k. This property constitutes one of the major assets of the method compared to the Gaussian mixture estimate: there exists a parameter which monotonically decreases the classification error. In the case of the Gaussian mixture estimate, the classification error reduction is generally obtained by increasing the number of Gaussian distributions per mixture. This property, however, relies on the training techniques implemented and performance cannot be guaranteed.

4.6 Forward-Backward Recursion

The previous estimation methods are based on classifiers. In this section, we present a different approach that is directly derived from the Baum-Welch algorithm for estimating the parameters of HMMs. As discussed in Section 3.2.2, this training algorithm relies on the state occupancy probability $\gamma(t, i)$, which is defined as

$$\gamma(t, i) = P(q_t = i|X, W) \tag{4.16}$$

where the variable q_t denotes the HMM states. These states typically represent a phoneme-level linguistic unit. The main advantage of this method is that the posterior probability is estimated using all the available acoustic information of the utterance, represented by the sequence of feature vectors X. Moreover, other sources of information can be introduced in the form of topological constraints such as the minimum duration of the phonetic units or multiple pronunciation transcriptions. In (Bourlard *et al.*, 2004), this method is investigated using a hierarchical approach.

Recently, posteriors estimated from the forward-backward recursion have been linearly combined with posteriors estimated from a MLP. These posterior features have then been used inputs for a tandem approach (Faria and Morgan, 2008).

4.7 Summary

In this chapter, we have presented a survey of ASR systems that use sub-word posterior probability estimates. Different estimation methods have been explored. Each method holds particular properties for obtaining more reliable posterior estimates. They involve multi-layer perceptrons, support

vector machines, maximum entropy models, Gaussian distributions, non-parametric techniques and forward-backward recursion. The common property of these estimation methods is that they all follow a discriminative procedure. In the case of GMM-based posteriors, a generative estimation is performed but a discriminative transformation is then applied to the posterior features.

Since using a MLP is the dominant method for estimating posterior probabilities, a large range of applications can be found in the literature. In this chapter, we have discussed the MLP-based posterior features depending on their role in the acoustic models:

Features : a vector formed by the posterior probability estimates is used as input features for a standard ASR acoustic model (typically HMM/GMM). This vector is generally transformed so that it can be better modeled by a GMM.

Scores : posterior probabilities are used as state scores in a HMM-based architecture. They are typically turned into scaled emission likelihood via Bayes' rule. Hence, the class posteriors must represent the same linguistic unit than the HMM states.

Labels : codewords are generated from the posteriors so that they can be used as inputs for discrete HMM. They are defined as the most probable posterior class. The discarded classes may contain information that is relevant for ASR.

Chapter 5

Template Matching Using Posterior Features

5.1 Introduction

Template matching (TM) for ASR relies on the assumption that the speech variability of each linguistic unit (e.g. words) can be captured through a set of samples (templates) obtained from the training data. A template in ASR is a sequence of feature vectors extracted from a particular pronunciation of a word. As a non-parametric approach, TM does not make any explicit assumption about the data distribution. Hence, it can potentially yield better performance than parametric HMM-based ASR approaches. Traditional TM use spectral-based speech features to characterize the templates and the test utterances. The main limitation of this approach is that a huge amount of templates is required to reliably describe all the possible different realizations of a word. Hence, in practice, this method is mainly oriented to small vocabulary and speaker-dependent recognition tasks.

In this chapter, we investigate the use of posterior features in the context of TM. We particularly study how the ASR performance is affected by the number of templates. Since posterior features can be seen as speaker-invariant speech features that are trained to convey the phonetic information of the speech signal (see Section 1.3), we can expect that these features are more stable and hence,

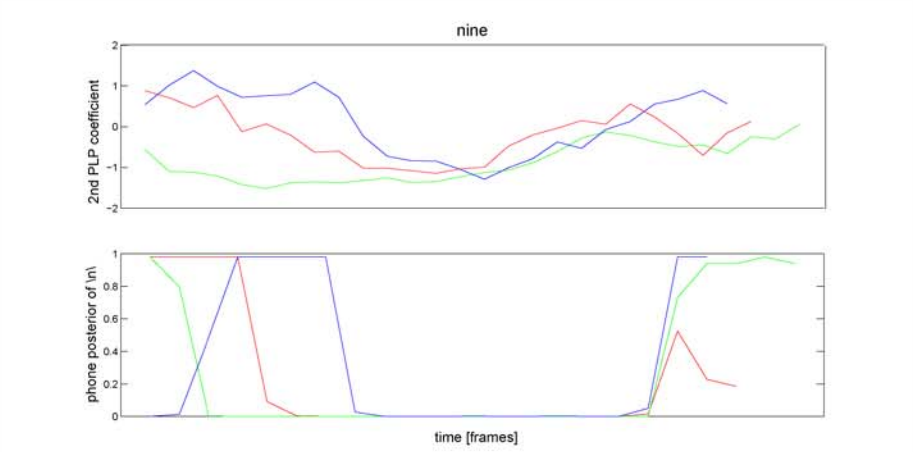

Figure 5.1. Temporal trajectory of one component of the feature vector using cepstral-based features and phoneme posteriors for three different samples of the word *nine*.

a limited number of templates can accurately describe a word. Figure 5.1 shows this effect. We can observe that the temporal trajectories of the spectral-based features follow different dynamics among them and are thus more difficult to describe. On the other hand, trajectories of posterior features are more stable and they follow a similar pattern. Hence, less templates will be required to characterize the dynamics of these features.

As described in Section 3.3, a local distance must defined to compute the similarity between the speech features forming the templates and the features characterizing the test utterance. Traditional local distances are based on the Euclidean distance. In this work, we also study the influence of other information theoretic measures that are more appropriate for describing the similarity between posteriors. In particular, we investigate the use of metrics derived from the KL divergence.

The experiments conducted in this chapter correspond to three different situations, depending on the size of the lexicon and the amount of training data:

- A large amount of training data is available and the size of the lexicon is small. This situation allows each word to be represented by a large number of samples. In this case, parametric models (i.e. HMMs) can represent either sub-word or whole word units because there are enough samples for describing both types of linguistic classes. The Digits database is used to simulate this situation.

- A large amount of training data is available and the size of the lexicon is also large. In this

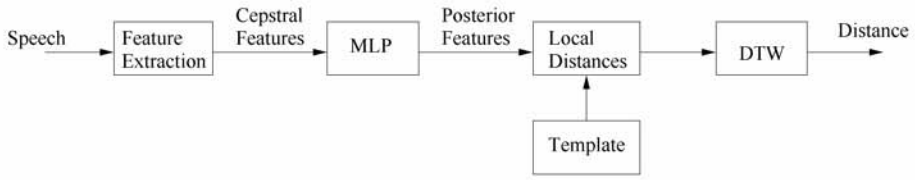

Figure 5.2. Block diagram of the TM-based approach for posterior features.

case, each word can only be represented by a few templates. Unlike the previous situation, HMMs can only model sub-word units so that they can obtain reliable estimates of their parameters. Word models are then obtained by concatenating sub-word HMMs using a phonetic transcription. The Resource Management database is used as an example of this situation.

- The amount of training data is limited. This situation typically appears in those applications, like voice-activated agendas, where the user describes the lexicon by providing a few acoustic samples and/or the grapheme transcription of each word. In this case, only a few templates can characterize each word and hence, training a parametric model using this database is not possible. State-of-the-art approaches use, in this case, an auxiliary database to learn the speech variability. In this work, we use the Phonebook and CTS databases to simulate this case.

The rest of the chapter is structured as follows: Section 5.2 presents an overview of the TM-based system for posterior features used in this work. Section 5.3 describes the local distances used in this work. Each local distance assumes some properties about the data that are also discussed. Section 5.4 describes the experimental setup of each of the different situations presented above. Finally, Section 5.5 summarizes and concludes this chapter.

5.2 System Overview

Figure 5.2 shows the main steps of the TM-based approached used in this chapter. Sub-word (phoneme) posterior probabilities are used as speech features to form the templates and the test utterances. These posteriors are obtained using cepstral-based speech features (PLP) as inputs to a MLP as described in Section 2.2.2. The MLP weights can be trained using the same training data where the templates are extracted from or they can be estimated from an auxiliary database. The

effect of using an auxiliary database for training the MLP will be explored in the experiments of this chapter.

A set of templates for each word in the vocabulary of the system is extracted from a training dataset. Each template is represented by a sequence of posterior features. Given a test utterance and a template, a matrix of local distances is computed between the posterior features forming the test utterance and the template. The DTW algorithm is then applied to this matrix to obtain the path that yields the minimum accumulated distance as described earlier in Section 3.3.2. This procedure is carried out for every template. The test utterance is finally recognized as the same word represented by template yielding the lowest distance. This is mathematically expressed in (3.29).

5.3 Local Similarity Measures

In this section, we describe the local measures for the DTW implementation (see Section 3.3) used in this work. We use the Euclidean distance because it is the typical local distance used in TM and we also present several KL-based measures that are applied when using posterior features.

5.3.1 Euclidean Distance

Euclidean distance is the most common similarity measure between features in a TM approach. This distance assumes that data follows a normal distribution since its definition is equivalent to the logarithm of the Gaussian function with identity covariance matrix[1] (see Section 3.3.3). Given two feature frames $\mathbf{a}$ and $\mathbf{b}$ of dimension K, Euclidean distance is defined as

$$d_{Eucl}(\mathbf{a},\mathbf{b}) = ||\mathbf{a}-\mathbf{b}||^2 = \sum_{i=1}^{K}(a_i - b_i)^2 \tag{5.1}$$

When considering the properties of posterior features (i.e., their components are non-negative and sum up to one), Euclidean distance does not appear to be a suitable similarity distance between two posterior features. The left-handed side of Figure 5.3 shows this effect. It plots the contour lines of the function defined by the Euclidean distance between the point defined by "a" to any other point

[1] Feature vectors are typically normalized in mean and variance to satisfy the condition of the identity covariance matrix.

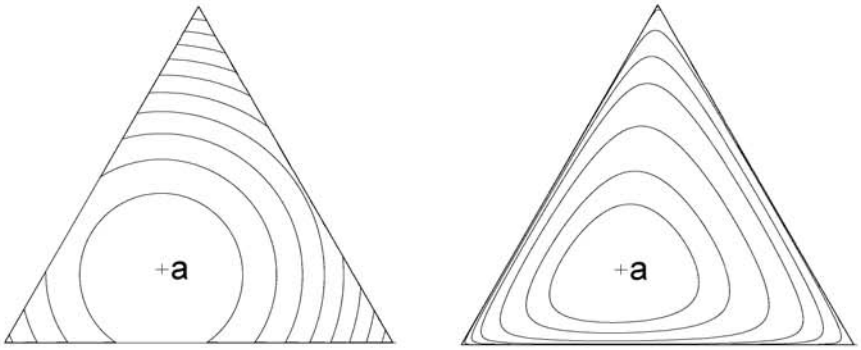

Figure 5.3. Contour plots for Euclidean distance and KL divergence on the simplex space generated by 3-dimensional posterior features. On the left side, the evaluated function is $f(\mathbf{b}) = ||\mathbf{b} - \mathbf{a}||^2$ whereas on the right side, the function is $f(\mathbf{b}) = KL(\mathbf{a}||\mathbf{b})$.

on the simplex-space generated by the 3-dimensional posterior features. As the Euclidean distance is a symmetric quadratic function, these contour lines are equidistant circumferences. From the figure, two observations can be made:

- Euclidean distance does not explicitly consider the inherent boundaries of the posterior space. This can be observed on the figure since the contour lines cross the space limits.
- The maximum distance is finite because the posterior space is bounded in terms of the Euclidean distance. In other words, the maximum distance of 2 occurs when the two vectors are delta distributions $\mathbf{a} = \delta_n$ and $\mathbf{b} = \delta_m$ centered at different points, $(n \neq m)$.

5.3.2 KL-based Measures

The KL divergence is generally considered as a natural distance between probability distributions (Cover and Thomas, 1991). Since posterior features can be seen as probability distributions over the space of classes, this measure can be used to compute the similarity between two posterior features. Given two discrete probability distributions $\mathbf{a}$ and $\mathbf{b}$ of dimension K, the KL divergence is defined as:

$$KL(\mathbf{a}||\mathbf{b}) = \sum_{i=1}^{K} a_i \log \frac{a_i}{b_i} \tag{5.2}$$

Although this function satisfies some good metric properties such as $KL(\mathrm{a}||\mathrm{a}) = 0$ and $KL(\mathrm{a}||\mathrm{b}) > 0$ when $\mathrm{a} \neq \mathrm{b}$, this measure cannot be considered as a distance according to the mathematical definition because it is not symmetric and it does not satisfy the triangular inequality property. As shown in Appendix A.1, distributions a and b play different and meaningful roles in the KL definition: a is considered the reference distribution whereas b is the measured distribution.

The contour plots of the right-handed side of Figure 5.3 illustrate some good properties of the KL divergence when compared to the Euclidean distance:

- KL divergence explicitly considers the boundaries of the posterior space. In fact, the input vectors of this measure must hold the properties of distributions (i.e. the components must be non-negative and sum up to one).

- The KL divergence between two distributions can be any non-negative value. Unlike in the case of the Euclidean distance, this value does not have an upper bound and can actually be infinite. This is due to the logarithm contained in the KL formulation.

These properties make KL divergence very appropriate as a similarity measure between posterior distributions. In fact, it is accepted as a natural distance between distributions (Cover and Thomas, 1991) and it can be shown that the space of distributions using the KL divergence as metric holds properties, such as Pythagoras' theorem, that are similar to the Euclidean spaces (Amari, 2001).

The use of KL divergence can be found in a large number of applications. In speech recognition, the Itakura-Saito distance (Nocerino *et al.*, 1985) was used as dissimilarity measure between two acoustic frames in a template matching approach. This distance can be interpreted as the KL divergence between two normalized speech power spectra. In speech synthesis, a symmetric version of KL is used as concatenation cost for unit selection (Klabbers and Veldhuis, 2001). KL divergence has also been effectively used in other fields. For information retrieval, it can be used to compute the similarity between the query and the document (Carpineto *et al.*, 2001). In this case, queries and documents are represented by normalized vectors of word frequencies. In the domain of DNA detection, a KL divergence has also been applied. In this case, nucleotide sequences are then described by their occurrence probabilities within sliding windows and compared to reference probability distributions estimated from a training dataset (Vinga and Almeida, 2003).

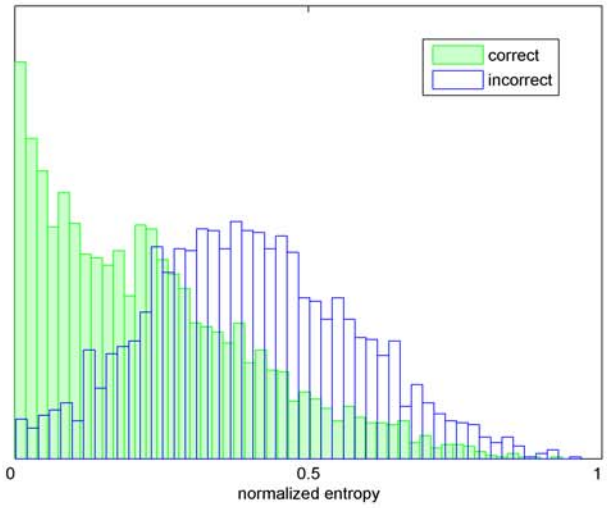

Figure 5.4. Histogram of correct and wrong posteriors according to their normalized entropy. The normalized entropy is defined as the entropy divided by the logarithm of the number of classes. Posteriors are considered correct if the class with the highest probability is correct, otherwise they are considered wrong.

Link between correct decision and entropy

In this section, we present an interesting property of the posterior features that will be exploited in the choice of the local distance. We empirically show that the classification error rate of posterior features is related to their entropy. As we explain in Appendix A.1, the entropy measures the degree of uncertainty provided by a probability distribution. Figure 5.4 plots the histograms of both correct and incorrect posterior features according to their entropy. In this case, the correctness of a posterior feature is evaluated depending on its highest component: if it corresponds to the target label is considered correct, otherwise is incorrect. From the figure, we can observe that the entropy of the correct posterior features is generally lower than the entropy of the incorrect posterior features. Hence, we can establish a correlation between the inverse entropy and the correctness of the posteriors. This relation is used in the next section, when we describe the different posterior-based local distances used in this chapter.

This link between the entropy and the correctness is quite intuitive: when the classification is correct, the posterior of the target class is close to one and hence, the entropy of the posterior feature is very low[2].

[2]A similar conclusion is reported in (Bourlard and Morgan, 1993), where the classification rate and the posterior value of the target class are compared.

The choice of the reference distribution

Given the asymmetric nature of the KL divergence, there exists two basic configurations when computing the similarity distance between posterior features. Let us consider z the frames corresponding to the test sequence and y the frames from the templates.

- $d_{KL}(\mathbf{z}, \mathbf{y}) = KL(\mathbf{y}||\mathbf{z})$. In this case, the frames belonging to the templates play the role of the reference distribution.

- $d_{RKL}(\mathbf{z}, \mathbf{y}) = KL(\mathbf{z}||\mathbf{y})$. In this case, posteriors features of the test sample are considered as reference distributions.

Since the local distance $d(\mathbf{z}, \mathbf{y})$ is always computed between a vector from the test sequence z and a vector of the template y, we can observe that d_{KL} naturally fits in the local distance definition because we are considering the vectors forming the template as the reference distribution.

In the previous definitions of local distances d_{KL} and d_{RKL}, the reference distribution is fixed and belonging to either the template or the test sequence. Based on the link between the inverse entropy and the classification correctness of the posterior features, we propose another local distance d_{weight} that is a weighted combination of d_{KL} and d_{RKL}.

$$d_{weight}(\mathbf{z}, \mathbf{y}) = \frac{w_1}{w_1 + w_2} d_{KL}(\mathbf{z}, \mathbf{y}) + \frac{w_2}{w_1 + w_2} d_{RKL}(\mathbf{z}, \mathbf{y}) \tag{5.3}$$

where $w_1 = \frac{1}{H(\mathbf{y})}$ and $w_2 = \frac{1}{H(\mathbf{z})}$. Thus, the proposed local distance considers both the posteriors from the template and the test utterance as reference distributions. The contribution of each weighting factor is related to the inverse entropy and hence, related to the correctness of the posterior feature as we have observed before. It can also be pointed out that d_{weight} is a symmetric measure.

5.4 Experiments and Results

As mentioned in the introduction, the experiments carried out in this chapter are divided into three categories. First, in Section 5.4.1 we evaluate the use of a large number of templates characterizing each word in a small vocabulary recognition task. We use, in this case, the Digits database. Then,

Section 5.4.2 considers the situation where few templates corresponding to each word in a large vocabulary recognition task are considered. In this case, the Resource Management database is used. Finally, we study in Section 5.4.3 the application where the amount of training data is very limited and hence, we need to use an auxiliary database to learn the speech variability. We use the Phonebook and the CTS database to evaluate this experiment.

Although templates can represent any linguistic unit from words to phoneme-level units, in this work, templates represent words in all the experiments. Using templates representing sub-word units can provide more accurate description of the speech variability of those linguistic classes because a larger number of templates would characterize each unit. However, given the large number of different units and templates per unit, pruning techniques are then required during the decoding stage for reducing the search space. This type of techniques are not within the scope of this thesis. A detailed description of this type of ASR systems can be found in (Wachter *et al.*, 2007) where the context (adjacent templates in the original training file) is also taken into account for further reducing the search space.

5.4.1 Digits Database

Experimental Setup

The structure of the Digits database allows that each word can be represented by a large number of templates. The influence of the number of templates on the accuracy of the system has been studied in all the experiments. Since the accuracy is highly dependent on the selected templates (especially when few of them are used), results correspond to the average performance of 3 experiments where each one has used a different template set. These template sets have been randomly chosen.

The effect of using an auxiliary database for training the MLP that estimates the posteriors is evaluated by using two MLPs. The weights of the first one are estimated using the training data of the Digit database whereas the second one is trained using the CTS database. Details about the structure of these two MLPs is shown in Table 2.1. In both situations, templates are obtained from the training data of the Digits database. The dimension of the posterior features corresponds, in each case, to the number of output units. Tandem features derived from posteriors as described in Section 4.1.1 are also evaluated.

Experiments using standard spectral-based speech features are also conducted for comparison. These features correspond to 39-dimension PLP vectors, i.e., the first and second order dynamic features are also included. Preliminary experiments using only static features (13 dimensions) and static plus first order dynamic features (26 dimensions) were also performed and lower accuracy was obtained.

Regarding the local distances, traditional Euclidean distance is compared to the KL-based measures described in the previous section. Euclidean distance is applied to both spectral and posterior-based features (including tandem) whereas the measures derived from the KL divergence are only applied to the posterior features. When using Euclidean distance, features are normalized in mean and variance so that Euclidean distance is equivalent to Mahalanobis distance.

The experiments carried out in this section are:

1. A comparison between posterior and cepstral features (Aradilla *et al.*, 2006b). Posteriors are extracted using the MLP trained on the Digits database. Also, a first comparison between the Euclidean distance and the KL divergence when using posterior features is performed.

2. The different posterior-based local distance described in the previous section are evaluated (Aradilla and Bourlard, 2007). In this case, results are shown for posteriors estimated from the MLPs trained on the Digits and CTS databases.

3. The best TM-based results are compared with state-of-the-art HMM/GMM. In particular, we use PLP and tandem features for this acoustic model. Tandem features are obtained from the posteriors estimated from the MLP trained on Digits. Moreover, we present the results using word and phoneme-level HMM/GMM. This is possible because the database contains a large number of samples for each possible word (digit).

Results

Results on the first experiment are shown in Figure 5.5. We can observe that the number of templates required to characterize each word are significantly fewer when using posteriors than when using PLP features. This is explained because, in theory, posterior features only convey phonetic information in a discriminative fashion, hence, only a few templates are needed to characterize each word. Moreover, we can note that the KL divergence yields better performance than Eu-

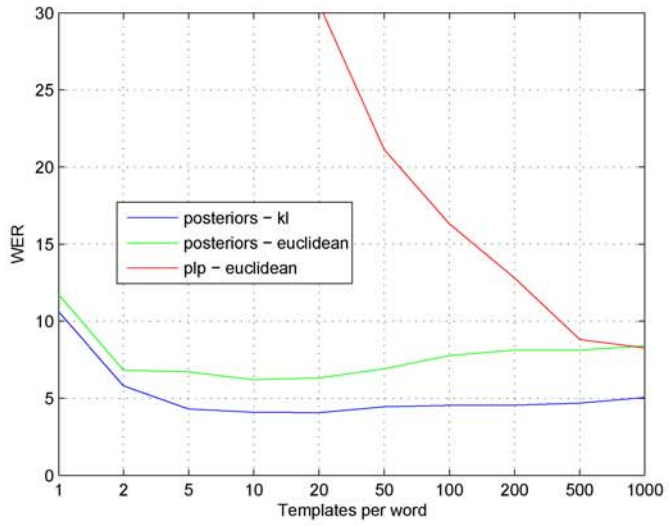

Figure 5.5. WER for the Digits database using a different number of templates per word. PLP and posterior features are used to form the templates and the test utterances. The KL divergence and the Euclidean distance have been applied when using posterior features. In the case of PLP features, only Euclidean distance can be used.

clidean distance. These results are consistent with the analysis on the contour plots in Section 5.3 that suggests that KL divergence is a more suitable measure for computing the similarity between posterior features than Euclidean distance.

Results on the second experiment are shown in Figure 5.6. In this experiment, two different MLPs are used to estimate the posterior features. The plots on the left side correspond to the MLP trained on the Digits database whereas the plots on the right side correspond to the MLP trained on the CTS database. Several observations can be made:

- When using posteriors estimated from the MLP trained on the Digits database, the accuracy slightly decreases after reaching a certain number of templates. Since the MLP has been trained on the same database, posterior features forming the templates are very reliable. Hence, only a small number of templates is required to characterize all the speech variability of the word. The contribution of additional templates is not relevant for providing new phonetic information but they can decrease the accuracy because of the posterior features containing errors due to mispronunciations or boundary misalignments when extracting the templates. This *over-training* effect is not observed when using tandem features because a large number of templates is then required to characterize the large value range of the trans-

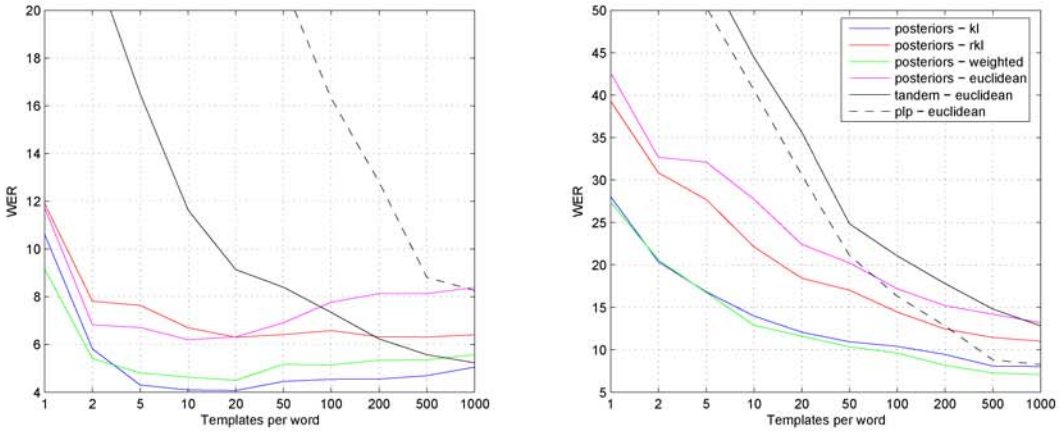

Figure 5.6. WER for the Digits database using a different number of templates per word. Two types of posterior features are used. The left-handed figure corresponds to the MLP trained on the Digits database whereas the right-handed figure corresponds to the MLP trained on the CTS database. It must be noted that the Y scale corresponding to the accuracy is different for the two plots.

formed posteriors. This can be explained by the non-linear transformation performed by the logarithm. This function has a high dynamic range for values close to zero, hence, the Euclidean distance of two posterior features with similar low components can yield to a high distance. The non-linearity appearing also in the KL divergence is compensated by the weighting factor of the reference distribution.

- Results obtained by the posteriors from the MLP trained on the CTS database are significantly inferior, especially when few templates are used. Given the mismatch between the data used for estimating the MLP parameters and test dataset, posterior features are less reliable and a larger number of templates is then required to characterize the speech variability of a word. However, posterior features from a MLP trained on a different database still outperform the performance of PLP features. These results will be exploited in the experiment described in Section 5.4.3 where the amount of data does not allow to train any parametric model.

- The local distances d_{KL} and d_{weight} yield better performance than the rest of the features. Although in this database, the difference between these two local distances is not significant, in the next experiments we will observe that d_{weight} can obtain a superior performance than d_{KL} especially when using very few templates. This effect is not observed in this database because results are already close to the maximum accuracy that can be obtained.

System	WER
TM - PLP (1000)	8.3
TM - posteriors (20)	4.0
TM - CTS posteriors (1000)	7.1
word HMM/GMM PLP	4.2
word HMM/GMM tandem	3.4
phoneme HMM/GMM PLP	4.1
phoneme HMM/GMM tandem	3.2

Table 5.1. Results using TM and HMM/GMM for the Digit database. A mixture of 16 Gaussian distributions is used to compute each state emission likelihood. The number between brackets denotes the number of templates per word.

- The ASR accuracy of the TM-based experiments using PLP and CTS-based posteriors constantly increasing when augmenting the number of templates. This suggests that a comparable performance could be obtained when using an even larger number of templates for characterizing each word. In fact, one of the properties of non-parametric approaches is that, given enough amount of training data, it can obtain twice the theoretical Bayesian error (Duda *et al.*, 2001). However, experiments using posteriors and KL-based local distances have shown that the appropriate choice of features and similarity measures can significantly decrease the amount of required templates.

The third experiment compares the performance of non-parametric (TM) and parametric (HMM/GMM) approaches. Table 5.1 reports these results. The input features of the HMM-based systems are posterior-based (tandem) and cepstrum-based (PLP) features. Also, HMM can represent, in this case, either context-dependent phonemes or whole words. We can observe that TM using posteriors can obtain a comparable performance to parametric approaches. However, these latter systems are more complex in terms of parameters and algorithms. Also, there is no difference in performance when using word or phoneme-level HMMs. Given the small vocabulary size of the database, the speech variability is equally well represented using either type of linguistic unit.

5.4.2 Resource Management Database

Experimental Setup

In this section, we evaluate the performance of the TM-based approach in a recognition task with a large lexicon size. Thus, the amount of templates characterizing each word cannot reach the size that is used in the Digit database. Again, besides the MLP trained on the same database, we

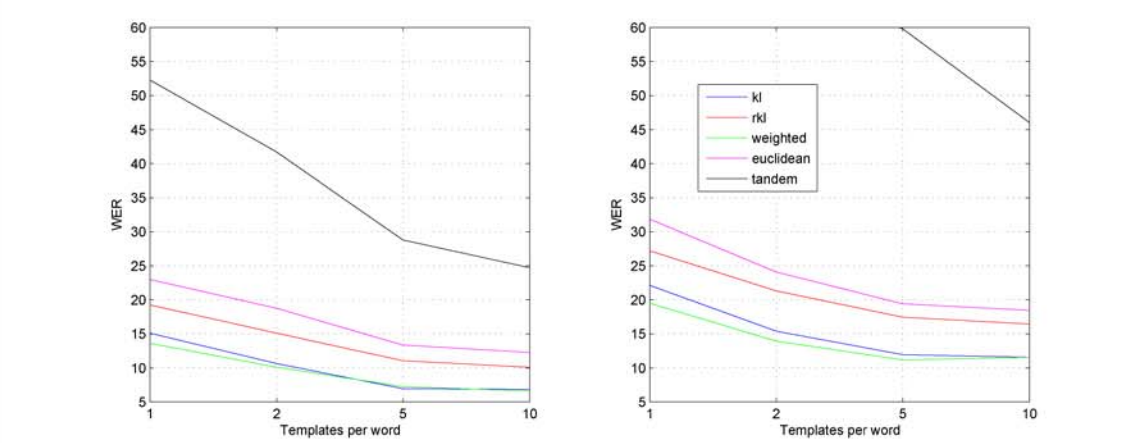

Figure 5.7. Results using TM for the RM database. Plots on the left-handed figure correspond to posteriors from the MLP trained on the RM database. Plots on the right hand correspond to the MLP trained on the WSJ database.

use a MLP trained on an auxiliary database (WSJ) to estimate the posterior features. Table 2.1 summarizes the information of both MLPs.

The experiments carried out in this section are similar to the second and third one of the previous section. First, we evaluate different local measures. Then, results obtained with the TM-based approach are compared with state-of-the-art HMM/GMM-based acoustic models. In this case, only phoneme-level HMM/GMM are used since the number of samples characterizing a word is not enough for estimating the parameters of the word-based HMMs.

Results

Figure 5.7 shows the results of the TM-based experiments varying the number of templates representing each word of the lexicon. Different local distances have also been studied and posteriors are estimated from the two different MLPs. As in the previous section, results using posteriors from the MLP trained on the same database are superior. Given the larger lexicon size, the number of templates per word is more limited than in the case of the Digits database. Hence, we do not observe the *over-training* effect when increasing the number of templates.

Regarding the local distances when using posterior features, we can observe that the difference in performances follows a similar behaviour to the results of the previous section: d_{KL} and d_{weight} perform better than the rest of distances and the Euclidean distance obtains the worse results since this measure does not explicitly considers the properties of posteriors. Although the performance of

d_{KL} and d_{weight} converge in accuracy when increasing the number of templates, the latter distance yields better performance when using very limited number of templates. This result is exploited in the experiment described in Section 5.4.3, where the structure of the database only allows two templates per word.

In addition, it can be pointed out that tandem features perform worse than the posterior features as it has been seen in the previous database. The trend indicated by the plots suggests that, given enough templates per word, tandem features could obtain a similar performance than the rest of local distances. Again, the large range of values that can take low posteriors requires the use of a larger number of templates for characterizing each word.

System	WER
TM - PLP (10)	52.3
TM - posteriors (10)	6.6
TM - WSJ posteriors (10)	11.1
phoneme HMM/GMM PLP	5.7
phoneme HMM/GMM tandem	5.7

Table 5.2. Results using TM and HMM/GMM for the RM database. State emission likelihoods use a mixture of 8 Gaussian distributions. The number between brackets denote the number of templates per word.

In a second experiment, we compare the performance of TM-based systems using the maximum number of templates with HMM/GMM-based acoustic models. Table 5.2 presents the results. We can observe that the performance obtained by PLP features is significantly low. A set of 10 templates is very low for characterizing the speech variability of a word using PLP features. Given the discriminative nature of posterior features, a limited number of templates is able to describe the variability of a word. As we observed in the previous database, the performance of the TM-based system when using posteriors from the MLP trained on the WSJ database is worse than when using posteriors from the MLP trained on the same database. However, its performance is significantly better than using traditional PLP features.

The performance of HMM-based systems is superior to the TM-based approach. The major different between these two systems is that HMM are modeling sub-word units (context-dependent phonemes in this case) and templates model whole words. Hence, the parameters estimated by the HMMs are more reliable because there are more training samples associated to each unit. A similar approach could be considered in the case of templates. Templates would represent sub-word units and thus, a large number of templates would characterize each unit (Wachter *et al.*, 2007).

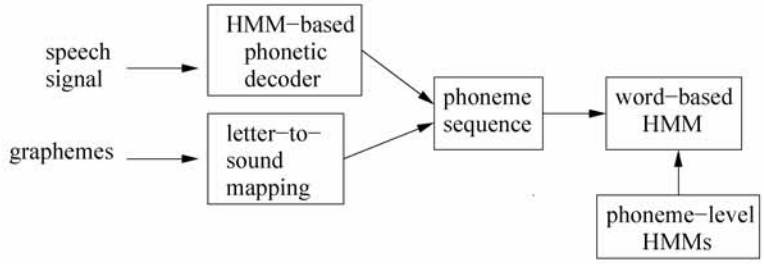

Figure 5.8. Block diagram for obtaining the word-based HMMs in state-of-the-art approaches. Two strategies can be applied to estimating the phonetic transcription of the lexicon depending whether the acoustic samples or the graphemes are provided.

5.4.3 Phonebook Database

The experiments in this section are thought to simulate an application, such as voice-activated agendas, where the user typically describes the lexicon of the system by providing a few acoustic samples or the grapheme (orthographic) transcription of each word. Obviously, in this case, there is not enough training data to model the speech variability required to perform a recognition task. Hence, this knowledge must be typically obtained from an auxiliary database. State-of-the-art systems use phoneme-level HMM/GMM models trained on a different database. The phonetic transcription of the word is obtained by applying a phonetic decoder to the acoustic samples provided by the user or by using a letter-to-sound mapping (Black *et al.*, 1998). This approach is represented in Figure 5.8. In this work, we use templates that are represented by a sequence of posterior features. These posteriors are estimated from a MLP that has been trained on an auxiliary database. In this case, we are exploiting the fact that posterior features can be estimated from a MLP trained on a different database, as we have observed in the TM-based experiments conducted in the previous databases. In this work, the MLP and the HMM/GMM models are trained on the CTS database.

The systems implemented in this work are mainly divided in three groups depending on if they use the acoustic information, the graphemes of the word or a combination of both (Aradilla *et al.*, 2008).

Using the Acoustic Information

In this work, we carry out experiments using one or two acoustic samples. This represents that the user has pronounced each word of the test vocabulary once or twice.

System 1 : This system implements TM using PLP features. This is the simplest system because

no information from other databases is considered and the acoustic samples provided by the user are directly used as references. In this case, PLP features are normalized in mean and variance and the Euclidean distance is applied.

System 2 : The word models of this system are based on HMM/GMMs representing context-dependent phonemes. The phonetic transcription required to form the word models is obtained by decoding the acoustic samples with an ergodic HMM using PLP features as input vectors. A phonetic transcription is then obtained from each acoustic sample. Hence, when two acoustic samples are used, each test word is described by two phonetic transcriptions. Context-dependent models are used for both obtaining the phonetic transcription of the acoustic sample and decoding the test utterances. This approach corresponds to the state-of-the-art in those applications where the lexicon is described by a few acoustic samples.

System 3 : This system implements the novel TM approach presented in this paper. Posterior features obtained from a MLP are used to form the templates and the test utterances. In this case, Euclidean distance and posterior-based measures described in Section 5.3 are used.

All the above systems use the acoustic information to build the word models. Systems 1 and 3 directly use the feature vectors extracted from the speech signal as a template and System 2 use the vector sequence to infer the phonetic transcription that will be used to form the HMM-based word model. Moreover, Systems 2 and 3 incorporate the information of the speech variability learned on the CTS database. While in System 2 this information is carried by the HMM/GMMs, in System 3 this information is applied by the MLP through the estimation of the posterior features.

The main difference between posterior-based templates (System 3) and HMM-based models (System 2) is that the former provides a confidence measure (posterior probabilities) of the phoneme set at each time frame. However, HMM-based models are based on the transcription of a phonetic decoder which outputs a single decision for each frame, i.e., the output will correspond to the most likely Viterbi path and no information is provided about the other paths.

Using the Grapheme Information

The grapheme information used in this work is provided by the Phonebook database. This information is used to infer the phonetic transcription of each word through a CART-based letter-to-sound

mapping (Breiman *et al.*, 1984). This technique is widely used in the speech synthesis field to obtain the phonetic transcription of a text (Taylor *et al.*, 1998). Phoneme-level HMMs trained on an auxiliary database can then be concatenated following the inferred phonetic transcription to build the system lexicon. Thus, each word is only represented by one phonetic transcription because there is only one grapheme transcription for each word. Since the test words of the Phonebook database are common English words, we can expect proper phonetic transcriptions.

System 4 : In this system, the phoneme-level models trained on an auxiliary database are based on HMM/GMM. This approach represent the state-of-the-art grapheme-based system in those applications where limited training data is available.

System 5 : In this system, the phoneme-level models are based on HMM/KL. This model is explained in detail in Chapter 6. It follows the structure as HMM/GMM but instead of state log-likelihood, the KL divergence is used to compute the state score. The interest of using this type of model in this work is that it can be directly combined with other systems using the KL divergence.

Using Grapheme and Acoustic Information

In this section, we describe a system that combines the information from both the grapheme and the acoustic samples.

System 6 : This system is a combination of System 3 and System 5. Each word in the lexicon is described by two models. A template from the acoustic sample and a HMM/KL obtained from the grapheme information. The score from DTW and from HMM/KL are combined to obtained the recognized word. The combination strategy is the minimum score. Hence, the decoding criterion expressed in (3.29) is extended so that it also includes the HMM/KL score

$$\text{class}(X) = \arg\min_{w \in \mathcal{W}} \min \left\{ \min_{Y \in \mathcal{Y}(w)} \varphi(X, Y), S(w) \right\} \tag{5.4}$$

where $S(w)$ is the score given by the HMM/KL model for word w.

Other combination strategies such as the sum have also been experimented. However, the combination based on the minimum score has shown to yield the best performance. This combination

		WER	
		1 sample	2 samples
System 1	d_{Eucl}	43.2	24.8
System 2		9.4	4.6
System 3	d_{Eucl}	18.5	11.6
	d_{KL}	8.6	4.3
	d_{RKL}	8.9	6.0
	d_{weight}	6.9	4.0
System 4		4.0	
System 5		5.3	
System 6	d_{weight}	3.6	2.8

Table 5.3. WER of the implemented systems. Systems using the acoustic information show two results corresponding to the use of one or two acoustic samples.

strategy has also been shown to be yield the best performance in other works, such as (Kuncheva, 2002).

System 6 benefits not only from the combination of two different information sources (grapheme and acoustic) but also from the complementarity of the acoustic models. The acoustic information is represented by templates while the grapheme information is used to build HMMs. The combination of HMM-based approaches with templates have been successfully applied for ASR (Axelrod and Maison, 2004; Aradilla *et al.*, 2005). These systems attempt to combine the good generalization capabilities of parametric models with the non-parametric approach provided by templates.

Results

Table 5.3 shows the results obtained by the systems described above. The following conclusions can be drawn:

- As expected, System 1 yields the lowest performance because it does not incorporate information about the speech variability learned from an auxiliary database. Hence, there is a limited capability to model the speech variability.

- Systems using the grapheme information generally yield a better accuracy than systems using the acoustic information. It must be noted that the phonetic transcription inferred from the grapheme transcription is particularly accurate because test words are common words. In a user-specific application, phonetic transcriptions would probably be more prone to errors and hence, it would yield a worse performance.

- The proposed method (System 3) outperforms the conventional TM approach (System 1). In addition, the use of KL-based local measures further improves the accuracy. In particular, d_{weight} yields significant improvement with respect to other measures because the contribution of d_{KL} and d_{RKL} depends on the entropy of each distribution. This result is consistent with the experiments conducted in the previous databases. Moreover, it can be observed that the accuracy of the proposed method when using d_{weight} is significantly better than state-of-the-art acoustic-based approach (System 2) and yields comparable results to state-of-the-art grapheme-based approach (System 4) when using two templates.

- The combination of the proposed method with HMM/KL further improves the performance of the system. This can be explained because both approaches are complementary in two ways. Firstly, word references are represented by two different type of models: templates and HMMs. Secondly, the information used to build these references is also different: templates are based on the acoustic information and HMM/KL is formed from the grapheme information.

5.5 Summary and Conclusion

In this chapter, we have studied the use of posterior features in the context of template-based ASR. Traditional approaches based on templates use speech features characterizing the short-term spectrum. The main limitation of traditional TM is that a huge number of templates is required to characterize all the possible pronunciations of a word. In this chapter we use posterior features to represent the templates and the test utterances. Based on the experiments conducted on different databases, the following conclusions can be drawn:

- When using posterior features, a significantly lower number of templates is needed to characterize the speech variability of a word when compared to traditional spectral-based speech features. This is explained because of the discriminative nature of posterior features, along with the facts that they can be trained to be speaker and environment-independent and, in theory, they only convey the phonetic information of the speech signal. These properties reduce the speech variability contained on the speech features and hence, fewer templates are needed to characterize a word.

- The use of the KL divergence has been shown to be an effective measure for computing the local distance between the posterior features forming the templates and the posterior features from the test utterance. The KL divergence has yielded significantly better results than traditional Euclidean distance because is an appropriate similarity measure between probability distributions and hence, it can explicitly consider the particular properties of this type of features.

- Given the asymmetry of the KL divergence, two basic configuration are possible: in the first one, posterior features from the template are the reference distribution whereas in the second configuration, posterior features from the test utterance play the reference role. Experiments have shown that the choice of the reference distribution in the KL divergence definition can significantly affect the ASR accuracy of the system. In particular, a weighted combination of the two basic approaches generally outperforms the rest of the local distances, especially when few templates are used. The weighting factors are, in this case, related to the inverse entropy of the posterior feature playing the role of the reference distribution.

- Posterior features can be estimated from a MLP trained on an auxiliary database. This property can be exploited in those applications where the amount of training data is limited. In this case, the few templates provided by the database can be represented by posterior features from a MLP trained on a large database. This approach has been shown to outperform state-of-the-art methods in this situation.

Chapter 6

Posterior-based HMM

6.1 Introduction

This chapter presents a novel acoustic model where sub-word posterior probabilities are directly used as input features. The topology of this model is similar to HMM/GMM (see Section 3.2.4). The main difference appears in the parameterization and matching function associated to each state. A state score is defined based on the Kullback-Leibler (KL) divergence between the posterior features and a multinomial distribution characterizing each state. Since posterior features can be seen as discrete distributions over the space of classes, KL divergence is an appropriate distance to describe the acoustic regions on the space of posterior features. In addition, the KL-based acoustic model presents some advantages with respect to the HMM-based models for posterior features:

- Since the matching function for each state is based on an appropriate measure for posteriors, no post-processing of the posterior features is required as it is the case of tandem features for HMM/GMM (see Section 4.1.1). Moreover, when compared to HMM/GMM much less parameters are required to characterize the acoustic region represented by the state.

- The components of the posterior features are not tied to the HMM states as in the case of hybrid HMM/MLP (see Section 3.2.5). Hence, a large number of models can be considered without changing the structure of the estimator of the posterior probabilities. In particular, context-dependent phonemes can be modeled through a MLP using output classes represent-

ing context-independent phonemes.

- Unlike discrete HMMs using a MLP as VQ (see Section 4.1.3), where posterior features are simplified so that only the component with the maximum value is considered, the proposed KL-based acoustic model is taking the whole posterior feature as input. It has been shown that including the information from all the classes can increase its generalization capabilities and hence, its ASR accuracy (Cerf *et al.*, 1994).

The rest of the chapter is structured as follows: Section 6.2 describes the KL-based acoustic model together with its algorithms for training and decoding. Section 6.3 presents an interpretation in terms of maximum likelihood estimation, information theoretic clustering or entropy of the state distributions depending on the KL-based measure used to compute the state score. Moreover, this KL-based acoustic model also provides a general framework where other acoustic models for posterior features can be seen as particular cases. In particular, section 6.4 discusses the links between hybrid HMM/MLP and discrete HMM with the KL-based acoustic model. Section 6.5 shows the results of the posterior-based acoustic models described in this chapter on several databases. In addition, a study on the complexity of the system is also presented. Finally, Section 6.6 summarizes and concludes this chapter.

6.2 KL-based HMM

In this section, the KL-based acoustic model together with its training and decoding algorithms are presented.

6.2.1 General Description

The KL-based acoustic model has the same topology as the HMMs described in Chapter 3. In this case, each state i is parameterized by a multinomial distribution $\mathbf{y}^i$ of the same dimension as the posterior features $\{\mathbf{z}_t\}$. The matching function for each state is based on the KL divergence between the posterior features and the state distribution of that state. The structure of this model is illustrated in Figure 6.1.

The components of the KL divergence $KL(\mathrm{a}||\mathrm{b})$ play different and meaningful roles: a is the

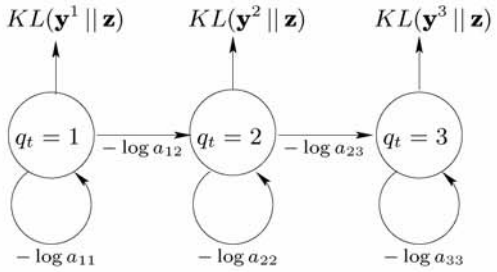

Figure 6.1. Scheme of a KL-based acoustic model formed by three states. The state score S is based on the KL divergence between the state distributions $\mathbf{y}^i$ and the posterior features $\mathbf{z}$. The state transition costs are defined as the negative log transition probabilities a_{ij}.

reference distribution and b is the distribution to be tested. Given this asymmetry, three different configurations can be investigated:

- HMM/KL: state distributions are the reference distribution: $S(\mathbf{y}^i, \mathbf{z}) = KL(\mathbf{y}^i, \mathbf{z})$
- HMM/RKL (reverse): posterior features are the reference: $S(\mathbf{y}^i, \mathbf{z}) = KL(\mathbf{z}, \mathbf{y}^i)$
- HMM/SKL (symmetric): KL and RKL are combined: $S(\mathbf{y}^i, \mathbf{z}) = \frac{1}{2}[KL(\mathbf{y}^i, \mathbf{z}) + KL(\mathbf{z}, \mathbf{y}^i)]$

The KL-based acoustic model can be seen as a prototype-based model (McDermott and Katagiri, 1994) where the prototypes are the state distributions. To the best of our knowledge, the presented model is the first attempt of applying prototype-based model to posterior features.

In the next sections, we present the algorithms for training and decoding the KL-based acoustic model. These procedures are based on the Viterbi approximation described in Chapter 3.

6.2.2 Training

The set $\theta = \left(\{\mathbf{y}^i\}_{i=1}^Q, \{a_{ij}\}_{i,j=1}^Q\right)$ denotes the parameters that describe the KL-based acoustic model. Each state i is parameterized by a multinomial distribution $\mathbf{y}^i$ and a_{ij} denote the transition probability from state i to j. The parameters θ are estimated by minimizing a cost function over the training data. This cost function is based on the KL divergence. Let us define a training set of N utterances where each utterance n is a sequence of $T(n)$ posterior features $Z(n)$. Let us also denote $M(n)$ the sequence of linguistic labels[1] corresponding to the training utterance n. The cost function

[1]In this work, experiments are carried out using context-independent and context-dependent phonemes but other types of linguistic units, such as words, can also be considered.

associated to the training utterance n is thus defined as

$$J_{\theta}^{M(n)}(Z(n)) = \min_{\mathcal{Q}(M(n))} \sum_{t=1}^{T(n)} \left[S(\mathbf{y}^{q_t}, \mathbf{z}_t(n)) - \log a_{q_{t-1}q_t} \right] \tag{6.1}$$

where $\mathcal{Q}(M(n))$ denotes the set of all possible state paths allowed by $M(n)$. The optimal parameters $\hat{\theta}$ can then be obtained by minimizing the sum of cost functions over all the training data.

$$\hat{\theta} = \arg\min_{\theta} \sum_{n=1}^{N} J_{\theta}^{M(n)}(Z(n)) \tag{6.2}$$

The algorithm presented in this thesis to obtain the optimal parameters $\hat{\theta}$ is similar to the embedded Viterbi algorithm for estimating the parameters of HMMs (Juang and Rabiner, 1990)[2]. It consists of two steps that are iteratively repeated until convergence of the parameters. In the first step, the optimal segmentation $\hat{\phi}(n)$ is obtained for each training utterance while keeping the state parameters θ fixed.

$$\hat{\phi}(n) = \arg\min_{\mathcal{Q}(M(n))} \sum_{t=1}^{T(n)} \left[S(\mathbf{y}^{q_t}, \mathbf{z}_t(n)) - \log a_{q_{t-1}q_t} \right] \tag{6.3}$$

The search over the space of state sequences $\mathcal{Q}(M(n))$ can be efficiently performed by using dynamic programming. In the second state, the segmentation is fixed and parameters θ are optimized.

$$\hat{\theta} = \arg\min_{\theta} \sum_{n=1}^{N} J_{\theta}^{\hat{\phi}(n)}(Z(n)) \tag{6.4}$$

This iterative procedure is guaranteed to converge to a local minimum because the state cost function defined by $S(\mathbf{y}, \mathbf{z})$ is a convex function (Juang and Rabiner, 1990).

From the point of view of the algorithmic interpretation, the first step (segmentation) obtains a mapping between states and all the posterior features in the training data. This mapping is obtained by doing forced-alignment between the linguistic units and the posterior features. In the second step (optimization), each state distribution $\mathbf{y}^i$ is computed given the set of $N(i)$ posterior features $Z(i)$ assigned to that state i.

[2]Appendix A.3 extends the re-estimation formulas obtained from the Viterbi algorithm towards the EM case, so that the state occupancy probability is taken into account.

Optimal State Distributions

- For the HMM/KL configuration, the optimal state distribution is defined as

$$\mathbf{y}^i = \arg\min_{\mathbf{y}} \sum_{\mathbf{z}\in Z(i)} KL(\mathbf{y}||\mathbf{z}) \tag{6.5}$$

  A closed solution can be obtained for this optimization problem:

$$y_k^i = \frac{\tilde{y}_k^i}{\sum_{k'=1}^{K} \tilde{y}_{k'}^i} \qquad \text{and} \qquad \tilde{y}_k^i = \sqrt[N(i)]{\prod_{\mathbf{z}\in Z(i)} z_k} \tag{6.6}$$

  where the subscript k denotes the dimension k. This solution represent the normalized geometric mean of the training posterior features.

- For the HMM/RKL model, state distributions follows a similar derivation

$$\mathbf{y}^i = \arg\min_{\mathbf{y}} \sum_{\mathbf{z}\in Z(i)} KL(\mathbf{z}||\mathbf{y}) \tag{6.7}$$

  and the new value of the state distribution $\mathbf{y}^i$ is

$$y_k^i = \frac{1}{N(i)} \sum_{z\in Z(i)} z_k \tag{6.8}$$

  This solution represents the arithmetic mean of the training posterior features.

- In the case of HMM/SKL, there is no closed solution to update the state distributions:

$$\mathbf{y}^i = \arg\min_{\mathbf{y}} \sum_{\mathbf{z}\in Z(i)} \frac{1}{2}[KL(\mathbf{y}||\mathbf{z}) + KL(\mathbf{z}||\mathbf{y})] \tag{6.9}$$

  However, the algorithm of an iterative procedure can be found in (Veldhuis, 2002) which only requires the arithmetic and geometric mean for obtaining the final solution.

The mathematical development of (6.6) and (6.8) is shown in Appendix A.2.

The transition probabilities a_{ij} are updated by counting and normalizing the transitions between states, as is done in standard Viterbi training.

6.2.3 Decoding

As in HMMs for ASR, the described model (although based on specific sub-word unit posteriors) can still be used to represent any linguistic unit, from words to context-dependent phonemes. Continuous speech recognition can be done by concatenating linguistic units, as in standard state-of-the-art ASR systems.

Given a sequence of posterior features Z of length T and a set of models $\mathcal{M}$ where each of them represents a different linguistic unit, a cost function or score $J_\theta^m(Z)$ is computed for each model m belonging to $\mathcal{M}$. The recognized linguistic unit $\hat{m}$ is the one with the lowest score

$$\hat{m} = \arg \min_{m \in \mathcal{M}} J^m(Z) \tag{6.10}$$

The model score $J^m(Z)$ is defined as

$$J_\theta^m(Z) = \min_{\mathcal{Q}(m)} \sum_{t=1}^{T} \left[S(\mathbf{y}^{q_t}, \mathbf{z}_t) - \log a_{q_{t-1} q_t} \right] \tag{6.11}$$

where $\mathcal{Q}(m)$ denotes the set of all the possible state sequences allowed by the linguistic unit m.

It should be noted that this decoding procedure is equivalent to the Viterbi decoding in standard ASR systems such as HMM/GMM. The minus log-likelihood estimated from the GMMs has been replaced, in this case, by a score based on the KL divergence between the state distributions $\{\mathbf{y}^i\}$ and the posterior features $\{\mathbf{z}_t\}$. Probabilistic grammars and word insertion penalties can then be used in a similar way as in HMM/GMM-based ASR systems.

6.3 Additional Interpretations

In this section, we further motivate the choice of the KL divergence. Besides being a natural dissimilarity measure between distributions, we show that HMM/KL can be also interpreted as a maximum likelihood model, HMM/RKL can be explained in terms of information theoretic clustering and the criterion of HMM/SKL assigns an appropriate entropy of the state distributions.

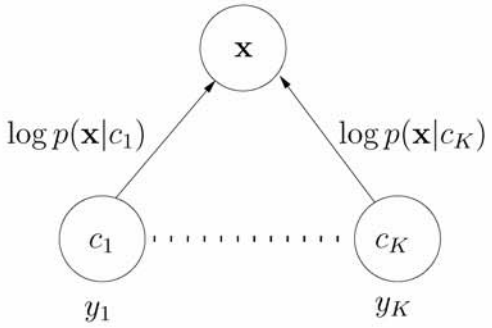

Figure 6.2. Each state i is described by a set of weights $\{y_k\}_{k=1}^K$. Each weight y_k corresponds to the class c_k which generates the conditional log-likelihood of the observation vector $\log p(\mathbf{x}|c_k)$.

6.3.1 HMM/KL

The training criterion for HMM/KL (6.5) is rewritten to have a deeper understanding of this model. In Appendix A.4, it is shown that (6.5) is equivalent to:

$$\arg\min_{\mathbf{y}} \sum_{\mathbf{z}\in Z(i)} KL(\mathbf{y}||\mathbf{z}) \approx \arg\max_{\mathbf{y}} \int_{\mathcal{X}(i)} p(\mathbf{x}) \sum_{k=1}^{K} [y_k \log p(\mathbf{x}|c_k)]\, d\mathbf{x} - KL(\mathbf{y}||P(\mathcal{C})) \tag{6.12}$$

where $P(\mathcal{C})$ denotes the multinomial distribution corresponding to the priors probabilities of the classes, i.e., $P(\mathcal{C}) = \{P(c_k)\}_{k=1}^K$ and $\mathcal{X}(i)$ defines the region on the acoustic space corresponding to the state i. The term corresponding to the integral can be directly compared with the expression obtained from a maximum likelihood estimation:

$$\hat{\Theta} = \arg\max_{\Theta} \frac{1}{N(i)} \sum_{\mathbf{x}\in\mathcal{X}(i)} \log p(\mathbf{x}^n|\Theta) \approx \int_{\mathcal{X}(i)} p(\mathbf{x}) \log p(\mathbf{x}|\Theta) d\mathbf{x} \tag{6.13}$$

The term within the integral $\sum_{k=1}^K [y_k \log p(\mathbf{x}|c_k)]$ can then be seen as the log-likelihood of a parametric model $\log p(\mathbf{x}|\Theta)$. This model is a convex combination of conditional log-emission probabilities $\log p(\mathbf{x}|c_k)$. Its parameters are the weighting factors $\{y_k\}_{k=1}^K$ of the convex combination. Each factor y_k determines the contribution of the log-emission probability of the acoustic vector x given the phoneme c_k. Hence, a meaningful interpretation can be obtained from the components y_k because they describe which classes are more representative for each state. Figure 6.2 shows a sketch of this structure.

The second term, $KL(\mathbf{y}||P(\mathcal{C}))$, in (6.12) compares the distribution obtained from the weighting

components y_k with the prior probability of the classes $P(\mathcal{C})$. If this term did not appear in the formula, the weighting factors would represent the trivial solution formed by one and zeros. On the other hand, if the integral term did not appear, the state distributions would be all the same and equal to the prior probability of the classes. Hence, the second term can be seen as a regularization factor.

6.3.2 HMM/RKL

In this section, the training criterion of HMM/RKL is reformulated in terms of information theoretic clustering. The optimal state distribution $\mathbf{y}^i$ for the HMM/RKL configuration expressed in (6.8) represents the arithmetic mean of the set of posterior features $Z(i)$ assigned to that state i. Hence, $\mathbf{y}^i$ can be seen as an estimator of the expected value of $Z(i)$. As described in Section 2.2.2, each posterior feature $\mathbf{z}$ represents, in fact, a multinomial distribution where each component is the posterior probability of a class c_k given an acoustic vector $\mathbf{x}$, i.e. $\mathbf{z} = P(\mathcal{C}|\mathbf{x}) = \{P(c_k|\mathbf{x})\}_{k=1}^{K}$. Assuming that event c_k is independent of the state i given the feature vector $\mathbf{x}$, the value defined by (6.8) can then be expressed as

$$\mathbf{y}^i = E_{\mathcal{X}(i)}[P(\mathcal{C}|\mathbf{x})] = P(\mathcal{C}|i) \tag{6.14}$$

Clustering is the unsupervised classification of patterns (features) into groups (clusters) that are represented by centroids. This mapping between features and centroids can be done using criteria based on information theory (Gokcay and Principe, 2002). The advantage of these criteria is that it is able to capture the data structure beyond second order statistics (variance). If we denote $\mathcal{X}$ as the set of all the acoustic vectors and $\hat{\mathcal{X}}$ the set of centroids of the acoustic vectors, it can be shown that (Tishby *et al.*, 1999)

$$I(\mathcal{X},\mathcal{C}) - I(\hat{\mathcal{X}},\mathcal{C}) \approx \sum_{i=1}^{Q} \int_{\mathcal{X}(i)} p(\mathbf{x}) KL(\underbrace{P(\mathcal{C}|\mathbf{x})}_{\mathbf{z}} \,||\, \underbrace{P(\mathcal{C}|i)}_{\mathbf{y}^i}) d\mathbf{x} \tag{6.15}$$

where I denotes the mutual information[3]. As it has been shown previously, the distributions $P(\mathcal{C}|\mathbf{x})$ and $P(\mathcal{C}|i)$ are respectively the posterior features $\mathbf{z}$ and the state distributions $\mathbf{y}^i$. The term in the

[3]See Appendix A.1 for an explanation of this measure.

right hand side is actually the criterion that is minimized for estimating the state distributions in the HMM/RKL configuration. The KL divergence is performed over all the training features and all the states as described by the sums over $\mathcal{X}(i)$ and i. Each state i is represented by $P(\mathcal{C}|i)$, which is the centroid of the set of posterior features $Z(i)$. Hence the minimization of (6.15) actually means finding those state distributions $P(\mathcal{C}|i)$, i.e., the set of centroids $\hat{\mathcal{X}}$, such that minimize the "loss of information" between the set of acoustic vectors $\mathcal{X}$ and the classes $\mathcal{C}$.

6.3.3 HMM/SKL

In this section, we empirically show the relation between the entropy of the training posterior features $H(\mathbf{z}_t)$ and the entropy of the state distributions $H(\mathbf{y}^i)$. These entropies can be computed following the standard definition (Shannon, 1948)

$$H(\mathbf{z_t}) = -\sum_k P(c_k|\mathbf{x}_t) \log P(c_k|\mathbf{x}_t) \tag{6.16}$$

$$H(\mathbf{y}^i) = -\sum_k y_k^i \log y_k^i \tag{6.17}$$

Following the same notation as in Section 6.2.2, we compute the average entropy $\bar{H}(i)$ of the training posterior features assigned to the state i as

$$\bar{H}(i) = \frac{1}{N(i)} \sum_{\mathbf{z} \in Z(i)} H(\mathbf{z}) \tag{6.18}$$

model	state entropy	MSE
HMM/KL	0.25	0.22
HMM/RKL	0.87	0.44
HMM/SKL	0.50	0.07
posteriors	0.51	-

Table 6.1. Columns indicate the average entropy of the state distributions and the average MSE between $H(\mathbf{y}^i)$ and $\bar{H}(i)$. The value on the last row is the average entropy of the training posterior features.

The second column of Table 6.1 shows the average entropy of the state distributions using context-dependent models. We can observe that, in the case of HMM/SKL, this value is very similar to the average entropy of the posterior features of the training dataset, shown in the last row of

the table[4]. The third column of Table 6.1 presents the average mean squared error (MSE) between $H(\mathbf{y}^i)$ and $\bar{H}(i)$. Again, we can observe that, when using the HMM/SKL criterion, the entropy of the state distributions is very similar to the average entropy of their assigned training samples. Figure 6.3 plots this correlation. Since the entropy is a measure of the uncertainty, this analysis shows that the criterion used in HMM/SKL is able to assign an uncertainty to each state distribution equivalent to the average uncertainty of their corresponding training samples.

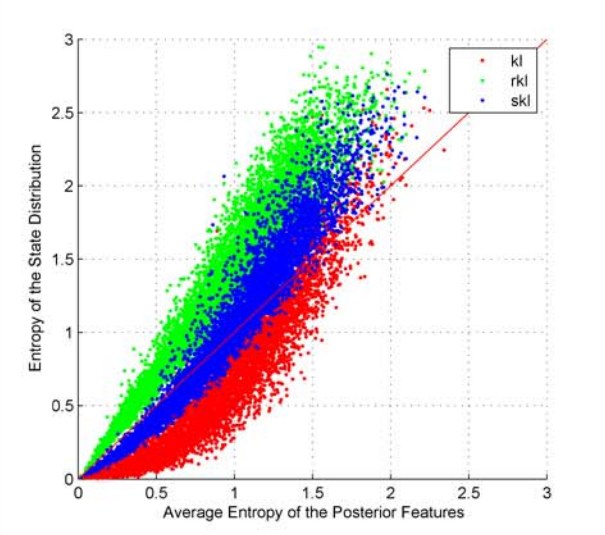

Figure 6.3. Each point corresponds to one state. The coordinates of each point represents the entropy of the state distribution and the average entropy of its assigned training posterior. These values are obtained from the context-dependent models.

It can be observed that symmetric KL is equivalent to the local distance d_{weight} (5.3) described in the previous chapter where the weights w_1 and w_2 are the same. This situation appears when the entropy of the reference distribution $H(\mathbf{y})$ is the same as the entropy of the posterior feature $\mathbf{z}$. On the other hand, we have observed that when using the criterion of HMM/SKL, the entropy of the training samples corresponding to the state i, $\bar{H}(i)$ is equivalent to the entropy of the state distributions. Hence, we can compare symmetric KL and d_{weight} by defining the weighting factors as

$$w_1 = \frac{1}{H(y^i)} \qquad \text{and} \qquad w_2 = \frac{1}{\bar{H}(i)} \tag{6.19}$$

[4]A similar average entropy is obtained from the test dataset.

Figure 6.4 plots the histogram of the weighting factor $w_1/(w_1 + w_2)$. We can observe that its value is close to 0.5 as in the state score definition of HMM/SKL. This property can be seen as extension of d_{weight} in the case of parametric models, where the state distribution is not fixed by the sequence of posterior features forming the templates but estimated from a training dataset.

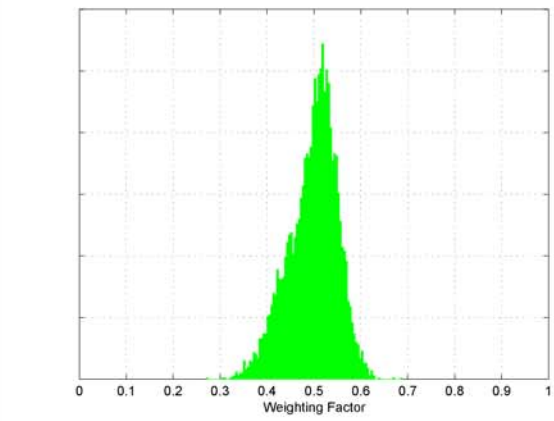

Figure 6.4. Histogram of the weighting factor $\frac{w_1}{w_1+w_2}$ when using CD models in the WSJ database.

6.4 Links with other Acoustic Models

In this section, we show that the proposed KL-based models can be considered as general cases of more standard acoustic models. Thus, we describe here the links that can be established between the KL-based model and some state-of-the-art acoustic models for posterior features. In particular, we discuss here the relationships between the proposed model with hybrid HMM/MLP, as well as discrete HMM.

6.4.1 Derivation of hybrid HMM/MLP

Expression (3.27) describes the acoustic score provided by hybrid HMM/MLP. We reproduce it here for the sake of clarity

$$\begin{aligned} J_H^m(Z) &= \max_{\mathcal{Q}^m} \left[\sum_{t=1}^{T} \log p(\mathbf{x}_t|q_t) + \log a_{q_{t-1}q_t} \right] \\ &= \max_{\mathcal{Q}^m} \left[\sum_{t=1}^{T} \log \frac{p(q_t|\mathbf{x}_t)p(\mathbf{x}_t)}{p(q_t)} + \log a_{q_{t-1}q_t} \right] \\ &\approx \max_{\mathcal{Q}^m} \left[\sum_{t=1}^{T} \log P(q_t|\mathbf{x}_t) + \log a_{q_{t-1}q_t} \right] \end{aligned} \tag{6.20}$$

where $\mathcal{Q}^m$ denotes the set of all possible state sequences allowed by the unit m and the prior probabilities of the classes are assumed to be uniform. In general, phoneme priors are similar among them except in the case of silence, which has a highest contribution than the rest. We conducted ASR experiments making this assumption and comparable performance to standard hybrid HMM/MLP was obtained.

When comparing (6.20) with the model score from HMM/KL (6.11), it can be noted that they are equivalent if we define the state distribution $\mathbf{y}^{q_t}$ as a delta distribution centered at the phoneme represented by the state q_t. Hence,

$$J_H^m(Z) = \max_{\mathcal{Q}(m)} \left[\sum_{t=1}^{T} -KL(\delta_{\rho(q_t)}||\mathbf{z}_t) + \log a_{q_{t-1}q_t} \right] \tag{6.21}$$

where $\rho(i)$ is the mapping from each state i to its corresponding class (MLP output). This represents the major limitation of hybrid HMM/MLP because each state must correspond to a class from the MLP. Then, a large set of classes, e.g., when using context-dependent phonemes, implies using a MLP with a large number of outputs, which is difficult to train because of the huge size of weights. HMM/KL removes this constraint because each state is only represented by a multinomial distribution whose values are estimated from the training data. Since state distributions are estimated from the training data, they can better model the variability of the posterior features, especially in the boundaries between phonemes, where the co-articulation effect is more pronounced. Figure 6.5 shows this effect. We can observe that state distributions from HMM/KL (Figures (b) and (c)) can better describe the trajectory of posterior features because they are estimated from a train-

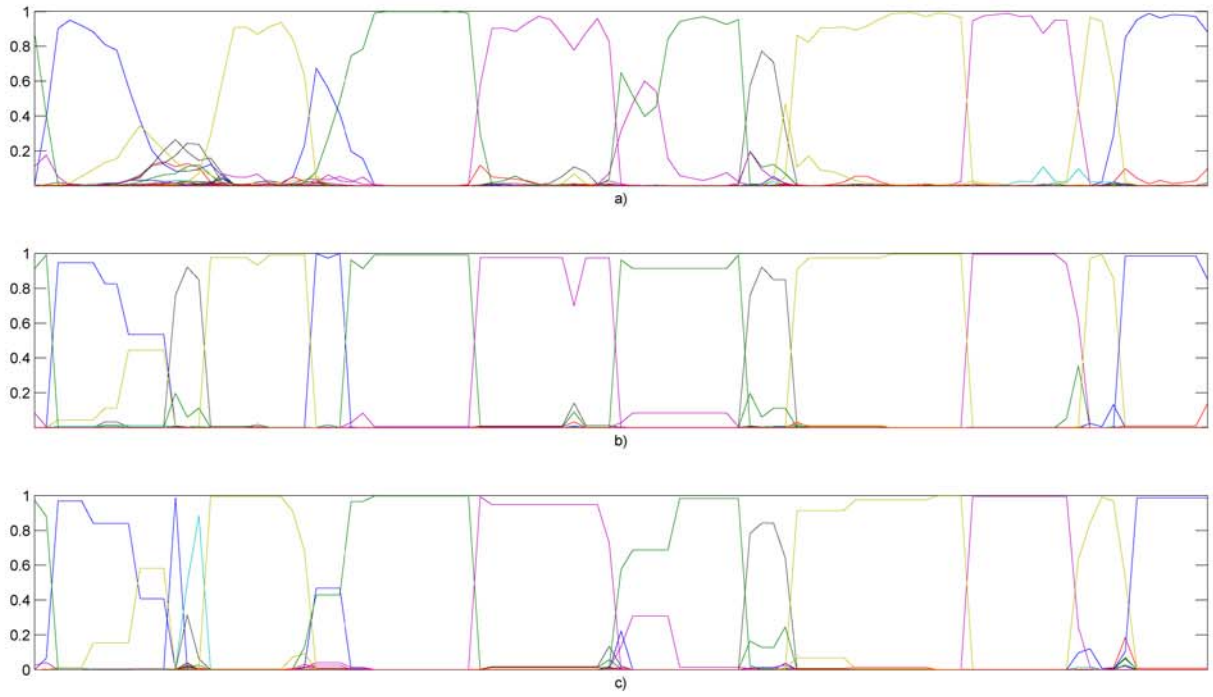

Figure 6.5. Three representations of the phoneme distributions in a spoken utterance along time. Figure (a) show the posterior features extracted from the MLP. Figures (b) and (c) show the multinomial distribution of the state assigned to each time frame corresponding to context-independent and context-dependent models respectively. In the case of hybrid HMM/MLP, these distributions are deltas.

ing dataset. In particular, context-dependent models in Figure (c) can explicitly model the influence of the adjacent models (co-articulation). Thus, the trajectory of state distributions is more similar to the posterior features in Figure (a).

Therefore, hybrid HMM/MLP can be seen as a particular case of HMM/KL where state distributions are delta distributions and phoneme priors are assumed uniform.

6.4.2 Derivation of discrete HMM

The acoustic score of a discrete HMM given a set of label indexes (codewords) $V = \{v_1, \cdots, v_T\}$ is defined as (3.28)

$$J_D^m(V) = \max_{\mathcal{Q}^m} \left[\sum_{t=1}^{T} \log P(v_t|q_t) + \log a_{q_{t-1}q_t} \right] \tag{6.22}$$

If we assume that the labels are obtained from a MLP, we can express the label sequence V as a sequence of delta distributions $\delta_{v_1}, \cdots, \delta_{v_t}, \cdots, \delta_{v_T}$, where each delta distribution δ_{v_t} is centered at the class with the highest probability assigned by the MLP, v_t. Thus the above expression can be rewritten as

$$J_D^m(V) = \max_{\mathcal{Q}^m} \left[\sum_{t=1}^{T} -KL(\delta_{v^t}||\mathbf{y}_{q_t}) + \log a_{q_{t-1}q_t} \right] \tag{6.23}$$

Therefore, discrete HMM can be seen as a particular case of HMM/RKL where posterior features are delta distributions centered at the component with the highest probability. It should be noted that this relation has already been observed in (Tsuboka and Nakahashi, 1994), where the discrete HMM was then using a fuzzy vector quantizer.

6.5 Results and Discussions

6.5.1 Experimental Results

Recognition experiments have been conducted on the Digits, Resource Management (RM) and Wall Street Journal (WSJ) databases. We compare the KL-based acoustic models described in this chapter with state-of-the-art HMM/GMM-based systems.

In this work, recognition experiments are based on phoneme-level units. Both context-independent (CI) and context-dependent (CD) models are considered. For the KL-based acoustic model, each unit is represented by a 3-state KL-based acoustic model. In the case of hybrid HMM/MLP, a minimum duration of 3 frames per phoneme is applied. When modeling CD phonemes, the most frequent 4000 triphones are used. This set represents 85% of the training data. The unseen triphones on the test set are modeled by the CI models. The initial state parameters are uniform distributions. Thus, the initial segmentation of the training dataset is also uniform[5]. Training iterations are stopped when the total cost function (6.1) on the training dataset does not decrease more than a given threshold. Evaluating the cost function on a cross-validation set yields to a similar number of iterations. This low over-fitting risk is due to the few parameters used to characterize each state. Moreover, training algorithms using Viterbi and EM procedures have yielded similar results. In this chapter, the KL-based models have been trained using the Viterbi algorithm.

For the HMM/GMM-based models, a mixture of 16 Gaussian distributions is used to estimate each state emission likelihood. In this case, CD phonemes are tied based on tree-based clustering using the log-likelihood criterion (Young *et al.*, 1994). Tandem features are obtained following the standard procedure described in (Hermansky *et al.*, 2000a).

Table 6.2 reports the results using CI and CD models. Since in this work, the MLP estimates posteriors of CI phonemes, hybrid HMM/MLP cannot use CD phonemes because each model corresponds to a MLP output. As we have mentioned in Section 2.6.4, some test utterances of the WSJ database contain out-of-vocabulary words (OOVs). We present results using the whole test set (913 utterances) and using the subset of test utterances does not contain OOVs (537 utterances).

6.5.2 Discussion

In this section, we study and discuss in detail the results reported in Table 6.2, mainly through the comparison between their related models.

[5]Other initial distributions, such as delta distributions from hybrid HMM/MLP, yield to similar parameter estimations.

model	Digits		RM		WSJ		WSJ (no-oov)	
	CI	CD	CI	CD	CI	CD	CI	CD
hybrid HMM/MLP	7.2	-	9.4	-	23.9	-	12.5	-
HMM/KL	7.8	6.0	9.0	6.2	23.5	22.3	11.8	11.5
discrete HMM	8.1	6.8	11.7	34.8	25.5	40.0	16.3	23.4
HMM/RKL	7.1	6.8	9.6	6.1	26.6	22.4	16.9	12.0
HMM/SKL	7.6	5.5	8.4	5.5	23.3	20.9	12.4	10.4
PLP - HMM/GMM	8.3	4.1	11.1	5.7	32.2	18.9	20.5	8.0
tandem - HMM/GMM	6.4	3.2	8.5	5.7	27.7	20.0	16.6	8.8

Table 6.2. WER on the Digits, RM and WSJ databases. CI and CD stand for context-independent and context-dependent models respectively.

Hybrid HMM/MLP and HMM/KL

The performance of hybrid HMM/MLP and HMM/KL are comparable when using CI phonemes. However, when modeling CD phonemes, we can observe a significant improvement over hybrid HMM/MLP. In this case, state distributions are able to model the co-articulation effects as illustrated in Figure 6.5. In Table 6.5.2, we can see an example of how state distributions can model the context. In particular, two CD phonemes (/*r*/-/*ih*/+/*k*/ and /*m*/-/*ih*/+/*l*/) corresponding to the same central phoneme (*ih*) are shown. We can observe that the left contexts (/*r*/ and /*m*/) are represented in the first state and that the right contexts (/*k*/ and /*l*/) appear in the third state. The second states correspond to the central phoneme (/*ih*/) that is also represented by the neutral vowel (/*ax*/).

/*r*/-/*ih*/+/*k*/		
1st	2nd	3rd
/*ih*/ (0.6)	/*ih*/ (0.9)	/*ih*/ (0.5)
/*r*/ (0.2)	/*ax*/ (0.1)	/*k*/ (0.4)

/*m*/-/*ih*/+/*l*/		
1st	2nd	3rd
/*ih*/ (0.8)	/*ih*/ (0.9)	/*l*/ (0.4)
/*m*/ (0.1)	/*ax*/ (0.1)	/*ih*/ (0.3)

Table 6.3. The two highest components of the 3 states distributions forming the triphone model are shown. The corresponding phoneme and its posterior value are represented.

MLP-based discrete HMM and HMM/RKL

Table 6.2 shows that, when using CI phonemes, discrete HMM compares to HMM/RKL. Although HMM/RKL constitutes a general case of discrete HMM, this latter model is more discriminant due to the null components appearing in some state distributions. They make state distributions more

discriminant because they cancel paths that are often wrong. In the case of HMM/RKL, there are no zero components in the state distributions because posterior features have not been simplified to delta distributions. This means that the state distributions of HMM/RKL are not discriminant enough. However, when using CD phonemes, HMM/RKL is able to significantly improve whereas discrete HMM decreases its performance dramatically. Since the number of models is higher, the amount of samples corresponding to each class is less and then, the number of zero components in the emission probabilities has increased for discrete HMM. Hence, some correct state paths are now canceled because the generalization capabilities of discrete HMM are not good enough.

Experiments have been carried out where delta distributions (labels from posteriors) have been used for the HMM/RKL models. These experiments have yielded slightly lower results than HMM/RKL using CD phonemes and significantly improves the performance of both discrete HMM and HMM/RKL using CI phonemes. Thus, we can take advantage of the good generalization properties of HMM/RKL during training and the fast decoding time of discrete HMMs.

HMM/SKL

HMM/SKL outperforms the rest of KL-based acoustic models as it was also observed in smaller databases (Aradilla *et al.*, 2007). This suggests that the entropy of the state distributions can significantly influence the performance of the system.

HMM/GMM-based models, using both posterior (tandem) and cepstral-based speech features (PLP) as inputs, significantly outperform KL-based acoustic models. As mentioned before, the main advantage of this type of models is that they can scale their number of parameters proportionally to the training data size by increasing the number of Gaussian distributions representing each state emission likelihood. This contrasts with KL-based acoustic models, where each state is only characterized by a single state distribution. Next section further evaluates the complexity of these systems.

6.5.3 System Complexity

In this section, we compare the performance of the acoustic models described in this paper using different number of parameters and training data sizes for the WSJ database. In particular, we want to evaluate if the increase of performance of HMM/SKL with respect to hybrid HMM/MLP is due

to the increase of parameters corresponding to the reference distributions. We are also interested in comparing the presented model with state-of-the-art HMM/GMM-based systems using the same number of parameters. Figure 6.6 shows the evolution of the performance using CD models and Table 6.4 presents the total number of parameters of the different models. The parameters of hybrid HMM/MLP are only the weights of the MLP. The hidden units of the MLP are chosen so that the total number of weights represents 5% of the training feature vectors. Experiments have empirically show that this ratio yields an optimal performance in large vocabulary recognition tasks (Ellis and Morgan, 1999).

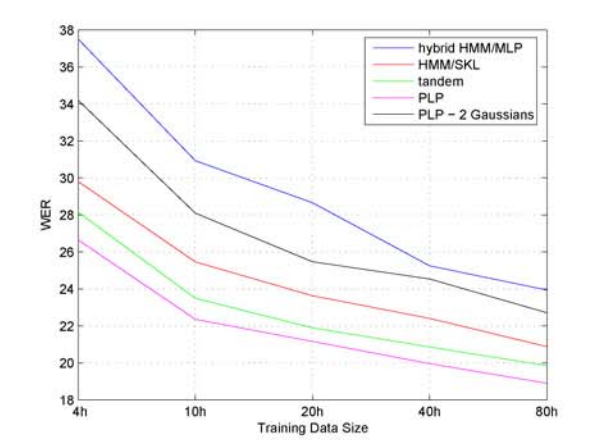

Figure 6.6. Word error rate depending on the training data size using CD models.

Table 6.4. Parameters of the acoustic models. The number of parameters of HMM/KL, HMM/RKL and HMM/SKL is the same.

Training	hybrid HMM/MLP	HMM/SKL	tandem	PLP
4h	72K	618K	1380K	1239K
10h	180K	726K	3571K	2391K
20h	361K	907K	6323K	4022K
40h	722K	1268K	9836K	6294K
80h	1446K	1992K	14328K	9382K

We can observe that all the models are similarly affected by the training data size. When comparing the proposed model HMM/SKL with hybrid HMM/MLP, it can be noted that hybrid HMM/MLP requires 80 hours of training data and a MLP of 1446K weights for obtaining the same performance as HMM/SKL, using only 20 hours of training data and 907K total parameters. This shows that characterizing each state by a reference distribution can be more efficient for modeling the speech variability than increasing the capacity of the MLP.

The number of parameters of the HMM/GMM-based system is superior to the KL-based models because we use 16 Gaussian distributions for modeling each state likelihood. A system using two Gaussian distributions per state for modeling the PLP features has a comparable complexity to HMM/SKL. Figure 6.6 shows that, in this case, the KL-based acoustic model yields significant better performance. In fact, experiments reveal that each state must be described by 6-8 Gaussian distributions to yield a comparable performance to HMM/SKL.

6.6 Summary and Conclusion

In this chapter, we have proposed a parametric model that uses posterior features directly as inputs. The presented model follows a similar topology than standard HMMs for ASR. A score function is associated to each state based on a KL-based measure between the posterior features and a multinomial distribution characterizing each state. The state multinomial distributions can be estimated by optimizing an objective function based on the KL divergence over a training dataset. Algorithms for training and decoding this KL-based acoustic model are also shown in this chapter. Since the KL divergence is not symmetric, several configurations have been studied:

- HMM/KL. In this case, the state multinomial distributions play the role of the reference distribution in the KL definition. By assuming uniform class priors, this configuration can be seen as a particular case of hybrid HMM/MLP where state distributions are delta distributions centered at the posterior class representing the same linguistic unit as the HMM state. Since state distributions are estimated from a training data and are not tied to the MLP outputs, context-dependent models can represent HMM states without changing the structure of the MLP.

  Results have shown that HMM/KL consistently improves the performance of hybrid HMM/MLP because of its capability of better describe the temporal evolution of the posterior features. Moreover, HMM/KL can be interpreted as a generative model where the components of each state distribution are estimated to maximize the log-likelihood of each state over a training dataset.

- HMM/RKL. In this case, posterior features play the role of the reference distribution in the

KL definition. Discrete HMM using a MLP for obtaining the codewords of the speech features can be seen as a particular case of HMM/RKL. Discrete HMMs are mainly chosen because of their fast decoding time since the computation of the emission likelihood consists of a simple table look-up. The main disadvantage of discrete HMMs is that labeling the speech features implies a binary reduction of the posterior features. Significant information may be lost in this process, especially when using context-dependent phonemes. HMM/RKL alleviates this limitation by considering the posterior features as soft boundaries. Experiments have shown that this generalization significantly improves the performance when comparing to discrete HMM. In this chapter, we also show that HMM/RKL acoustic models can be trained using posterior features and decoding can be done by using hard posteriors (labels). Thus, the posterior features can provide enough generalization during training and decoding can be simply performed by a look-up table.

In this chapter, we also show that the training criterion of HMM/RKL actually corresponds to a minimization of the loss of information between the acoustic features and the centroids represented by the states.

- HMM/SKL. In this case, a symmetric combination of the KL and RKL is used as state score. This model outperforms the previous KL-based models, HMM/KL and HMM/RKL. This can be explained because the symmetric KL criterion assigns an appropriate entropy to the estimated state multinomial distributions. Analysis have shown that the average entropy of the training posterior features assigned to a given state is equivalent to the entropy of the multinomial distribution representing that state.

  When comparing to state-of-the-art HMM/GMM-based systems, KL-based acoustic models yield worse performance when using CD models. This can be explained because HMM/GMM-based systems can increase the complexity of the systems by adding more Gaussian distributions for estimating the state emission likelihood. However, states in KL-based acoustic models are only represented by a single reference distribution. Experiments using an equivalent number of parameters have shown that HMM/SKL can outperform HMM/GMM-based models. Moreover, experiments comparing hybrid HMM/MLP and HMM/SKL have shown that representing each state by a reference distribution is more efficient for ASR than increasing

the capacity of the MLP.

Chapter 7

Summary and Conclusion

In this thesis, we have investigated novel acoustic models for ASR that can directly use posterior probabilities of sub-word units as input speech features. These posteriors can then be seen as speech features holding very convenient properties for ASR, such as being discriminative and speaker-invariant. Although in this work, we have used phonemes as posterior classes, other types of sub-word units can also be considered.

A critical issue in ASR is the definition of the similarity measure between the feature vectors extracted from the test utterance and the parameters characterizing the acoustic model. Traditionally, this similarity measure is based on general-purpose metrics, such as Euclidean distance or likelihoods computed from a mixture of Gaussian distributions. Using such similarity measures, the particular properties of posterior-based features are not fully considered. Moreover, in order to make posterior features suitable with traditional similarity measures, they are often processed by using a specific "ad-hoc" transformation that further dilutes the convenient properties of posterior-based speech features.

In this work, we have exploited the fact that posterior features can be seen as discrete distributions over the space of classes. We have then proposed the use of the KL divergence as similarity measure within ASR acoustic modeling. More specifically, we have applied this measure in the two main acoustic model paradigms currently used in the ASR field: non-parametric methods using templates and parametric models relying on the HMM architecture.

7.1 Template Matching Using Posterior Features

Phoneme posterior probabilities are used as speech features to form the templates and the test utterances. Based on experiments conducted on different ASR tasks, the following conclusions can be drawn:

- Given the discriminative capabilities of posterior features, a reduced number of templates can properly characterize the speech variability contained in a word. This is due to the fact that, in theory and to a large extent also in practice, posterior features only contain information about the phonetic content of the speech signal. Thus, information related to the speaker and environment has been removed and consequently fewer templates are necessary to characterize a word.

- Measures related to the KL divergence can yield significantly better accuracy than traditional Euclidean-based local distances. KL divergence explicitly considers the space topology of posterior features and hence, it can better describe the similarity between posterior features. Moreover, the uncertainty of the phonetic classification represented by a posterior feature can be characterized by its entropy. This measure can then be used as a weighting factor to further improve the accuracy of the system.

- The MLP used for estimating the posterior features can be trained on an auxiliary database. This approach is shown to be very effective in those ASR applications where the amount of training data is very limited. The few acoustic samples provided by the limited database are represented by a sequence of posteriors features estimated from a MLP trained on large database.

7.2 Posterior-based HMM

We generalized the posterior-based template matching framework by defining a HMM-like parametric model where each state is parameterized by a multinomial distribution and the state likelihood is replaced by a similarity measure based on the KL divergence. Since the KL divergence is not symmetric, three configurations are studied:

1. HMM/KL: state distributions are considered the reference distributions. Hybrid HMM/MLP can be seen as a particular case of this configuration when phoneme priors are uniform and state distributions are delta distributions centered at the phoneme represented by the state. Unlike hybrid HMM/MLP, this approach does not require modifications to the MLP structure for modeling context-dependent phonemes. Moreover, this model can be interpreted in terms of maximum likelihood terms. Results on different databases show that HMM/KL can outperform hybrid HMM/MLP because of its ability to model the co-articulation effects through context-dependent phonemes.

2. HMM/RKL: posterior features represent the reference distributions. This configuration can be seen as a generalization of discrete HMM where all the components of the posterior features are considered instead of only the most probable class. This generalization yields improvement in the word accuracy. Moreover, the fast decoding time of discrete HMMs can be also obtained when using labels on HMM/RKL models. In this case, the lack of generalization capabilities when using a large number of models (e.g., context-dependent models) is successfully handled because all the components of the posterior features are used for training. HMM/RKL can also be interpreted in terms of information-theoretic clustering.

3. HMM/SKL: a symmetrical combination of KL and RKL is used as state likelihood. This configuration outperforms HMM/KL and HMM/RKL. This can be explained because HMM/SKL is able to assign an appropriate entropy to the state distribution based on the average entropy of its training posterior features.

In comparison to state-of-the-art HMM/GMM-based systems, KL-based acoustic models yield worse performance when using CD models. This can be explained because HMM/GMM-based systems can take better advantage of large quantities of data by increasing the model complexity with a larger number of Gaussian components per state. In contrast, states in KL-based acoustic models are only represented by a single reference distribution per state. Experiments using an equivalent number of parameters have shown that HMM/SKL can outperform HMM/GMM-based models. Moreover, experiments comparing hybrid HMM/MLP and HMM/SKL have shown that representing each state by a reference distribution is more efficient for ASR than increasing the capacity of the MLP.

7.3 Future Directions

There are different research directions that could be followed in the future, including:

- Defining criteria for selecting the most informative templates. This would reduce computational time and would increase the system accuracy by removing noisy templates that can lead to classification errors. This approach, though, would certainly diverge from the template matching philosophy of using all training data without making any assumptions about its structure.

- Increasing the capacity of the KL-based acoustic models by characterizing each state with several state distributions. This approach would be similar to state-of-the-art HMM/GMM systems where each state is characterized by a mixture of Gaussian distributions. The entropy of the state distributions can be investigated as a measure of the variance of each mixture component. This approach can probably be linked with the criteria for template selection in the previous research direction.

- Both template matching and the KL-based acoustic models can be improved by using a larger set of posterior classes. Thus, posterior features will lie on a high dimensional space. This will also result in an increase in capacity of the KL-based models. A possibility for increasing the number of classes can be done by combining different posterior estimators (e.g. posteriors from the forward-backward recursion and MLP) or by combining posteriors of different classes (e.g. articulatory features). The investigation of strategies for combining different posterior estimators is also an interesting research direction.

Appendix A

Appendixes

A.1 Elements of Information Theory

Entropy

Let us first consider an information source generating independent labels from a set $\mathcal{A} = \{1, \cdots, m, \cdots, M\}$ as shown in Figure A.1. In this case, the output of the source can be seen as a random variable A described by the probability distribution $\{p(A = m)\}$.

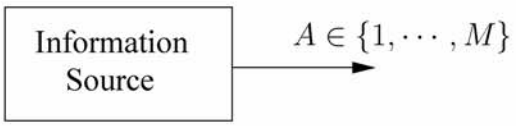

Figure A.1. A discrete memoryless information source.

For coding the output of this source, we need to assign a binary sequence to each possible label. Let us define l_m the length of the binary code corresponding to the label m. We can then compute the average code length $\bar{l}$ of the information source generating A

$$\bar{l} = \sum_{m=1}^{M} = p(A = m) l_m \tag{A.1}$$

It can be shown that minimum number of average bits $\bar{l}$ is reached when the number of bits assigned to each label a_m is related to its output probability $p(a_m)$ using the following formula-

tion (Shannon, 1948)

$$l_m = -\log p(a_m) \tag{A.2}$$

We can then define a measure called *entropy* $H(A)$ as

$$H(A) = -\sum_{m=1}^{M} p(a_m) \log p(a_m) \tag{A.3}$$

which corresponds to the minimum number of average bits required for coding an information source described by the output probabilities $\{p(A = m)\}$.

The *entropy* can also be seen as a measure of uncertainty of the information source. Let us consider the situation where $p(A = k) = 1$ for a given label k and the rest of output probabilities are zero. In this case, the output of the information source is completely deterministic because it is always k. Hence, the uncertainty is zero. The entropy of the source is also, in this case, equal to zero. On the other hand, if all the output probabilities are equal, i.e. $p(A = m) = 1/M$, the entropy and the uncertainty are maximum and equal to $\log M$.

Kullback-Leibler Divergence

Let us now consider that the information source is coded using a bit assignment that is optimal for source described by a different probability distribution $\{q(A = m)\}$. In this case, the average number of bits $\bar{l}_q$ is

$$\bar{l}_q = -\sum_{m=1}^{M} p(A = m) \log q(A = m) \tag{A.4}$$

and $\bar{l}_q \geq H(A)$ because the bit assignment $\{-\log q(A = m)\}$ is not optimal. We can then define a non-negative measure as $\bar{l}_q - H(\mathcal{A})$. This difference is defined as the *Kullback-Leibler (KL) divergence*

$$\bar{L} - H(A) = KL(p||q) = \sum_{m=1}^{M} p(a_m) \log \frac{p(a_m)}{q(a_m)} \tag{A.5}$$

This measure arises naturally in a wide range of problems such as hypothesis testing (Kem-

perman, 1967) or rate distortion theory (Berger, 1971). It can be considered as a natural distance between distributions (Cover and Thomas, 1991) and, in fact, using the KL divergence, we can establish parallelisms between the space of the probability distributions and the Euclidean geometry (Amari, 2001).

Mutual Information

Using the concept of entropy, we can define another measure that expresses the mutual dependency of two random variables. Given two random variables A and B, the mutual information $I(A, B)$ is defined as

$$I(A,B) = H(A) - H(A|B) \tag{A.6}$$

$$= H(B) - H(B|A) \tag{A.7}$$

$$= H(A) + H(B) - H(A,B) \tag{A.8}$$

Intuitively, if entropy $H(A)$ is regarded as a measure of uncertainty about a random variable, then $H(A|B)$ is a measure of what B does not say about A. This is "the amount of uncertainty remaining about A after B is known". This measure can also be expressed as the KL divergence between the joint probability between A and B, $P(A, B)$ and the product of their marginal probabilities, $P(A)$ and $P(B)$. Hence, it is a non-negative measure that is only zero when the random variables A and B are independent.

A.2 Optimal State Distributions for HMM/KL and HMM/RKL

HMM/KL Model

The problem is defined as follows: given a set of $N(i)$ posterior features $Z(i)$ assigned to the state i, find $\mathbf{y}^i$ such that minimizes

$$f(\mathbf{y}^i) = \sum_{\mathbf{z} \in Z(i)} KL(\mathbf{y}^i || \mathbf{z}) \tag{A.9}$$

Since $\mathbf{y}^i$ is a multinomial distribution, the constraint $\sum_{k=1}^{K} y_k^i = 1$ must be satisfied. We use the method of Lagrange's multipliers to find the solution:

$$\frac{\partial}{\partial y_k^i} \sum_{\mathbf{z} \in Z(i)} \left[KL(\mathbf{y}^i \,||\, \mathbf{z}) + \lambda \left(\sum_{k=1}^{K} y_k^i - 1 \right) \right] = 0 \tag{A.10}$$

Developing the above expression yields to:

$$N(i) \left(\log y_k^i + 1 \right) - \sum_{\mathbf{z} \in Z(i)} \log z_k + \lambda = 0 \tag{A.11}$$

$$\log y_k^i = \frac{\sum_{\mathbf{z} \in Z(i)} \log z_k}{N(i)} + \underbrace{\frac{\lambda}{N(i)} + 1}_{\lambda'} \tag{A.12}$$

$$y_k^i = \lambda' \sqrt[N(i)]{\prod_{\mathbf{z} \in Z(i)} z_k} \tag{A.13}$$

where λ' is the normalization factor that satisfies the constraint. Therefore, $1/\lambda' = \sum_{k'=1}^{K} \sqrt[N(i)]{\prod_{\mathbf{z} \in Z(i)} z_{k'}}$.

HMM/RKL Model

Similarly to the KL case, the function to be minimized is then defined as:

$$f(\mathbf{y}^i) = \sum_{\mathbf{z} \in Z(i)} KL(\mathbf{z} || \mathbf{y}^i) \tag{A.14}$$

Applying Lagrange's multipliers, we obtain that

$$\frac{\partial}{\partial y_k^i} \sum_{\mathbf{z} \in Z(i)} \left[KL(\mathbf{z} || \mathbf{y}^i) + \lambda \left(\sum_{k=1}^{K} y_k^i - 1 \right) \right] = 0 \tag{A.15}$$

$$\sum_{\mathbf{z} \in Z(i)} \frac{z_k}{y_k^i} + \lambda = 0 \tag{A.16}$$

$$y_k^i = \frac{-1}{\lambda} \sum_{\mathbf{z} \in Z(i)} z_k \tag{A.17}$$

In order to guarantee that $\sum_k y_k^i = 1$, $\lambda = -N(i)$. Hence,

$$y_k^i = \frac{1}{N(i)} \sum_{\mathbf{z} \in Z(i)} z_k \tag{A.18}$$

It can be noted that the optimal distribution for the KL case is the normalized geometric mean of the set $Z(i)$, whereas for the RKL configuration, the optimal distribution is the arithmetic mean.

A.3 EM-based Re-estimation of State Distributions

In this section, we develop the formulas for re-estimating the state distributions of the KL-based parametric models using the EM algorithm. As we have seen in Section 3.2.2, this training algorithm weights each training sample $\mathbf{x}_t$ according to the posterior probability of the hidden variable $q_t = i$. This posterior probability can be computed from the alpha and beta recursions as shown in (3.10), (3.11) and (3.12).

$$\alpha(t,i) = \sum_j \alpha(t-1,j) a_{ji} b_i(\mathbf{x}_t) \tag{A.19}$$

$$\beta(t,i) = \sum_j \beta(t+1,j) a_{ij} b_j(\mathbf{x}_{t+1}) \tag{A.20}$$

$$\gamma(t,i) = P(q_t = i|X,W) = \frac{\alpha(t,i)\beta(t,i)}{\sum_j \alpha(t,j)\beta(t,j)} \tag{A.21}$$

In the context of the KL-based model, we define the emission distribution of the posterior feature $\mathbf{z}_t$ given the state i as

$$b_i(\mathbf{z}_t) = \exp\{-S(\mathbf{y}^i, \mathbf{z}_t)\} \tag{A.22}$$

Thus, we can compute the alpha, beta and gamma probabilities as described above.

In the following, we derived the re-estimation formulas presented in the previous section so that the state occupancy $\gamma(t,i)$ is taken into account. Although these formulas are computed for a single training utterance, they can be extended to a set of training utterances by simply summing over all the training dataset.

HMM/KL Model

Equation A.13 can be rewritten in the logarithm domain as

$$\log y_k^i = \log \lambda' + \frac{\sum_{\mathbf{z} \in Z(i)} \log z_k}{N(i)} \tag{A.23}$$

By considering a single training utterance of T posterior features and the indicator function $I(A)$ that outputs one if A is true and zero otherwise, we can rewrite the above expression as

$$\log y_k^i = \log \lambda' + \frac{\sum_{t=1}^{T} I(\mathbf{z}_t \in i) \log z_{tk}}{\sum_{t=1}^{T} I(\mathbf{z}_t \in i)} \tag{A.24}$$

where z_{tk} denotes the kth component of the posterior feature $\mathbf{z}_t$. Then, we can substitute the indicator function $I(\mathbf{z}_t \in i)$ by its probabilistic version, the state occupancy $\gamma(t, i)$

$$\log y_k^i = \log \lambda' + \frac{\sum_{t=1}^{T} \gamma(t, i) \log z_{tk}}{\sum_{t=1}^{T} \gamma(t, i)} \tag{A.25}$$

where λ' is the normalization factor.

HMM/RKL Model

In a similar way as in the case of the HMM/KL model, we can rewrite the re-estimation formula for HMM/RKL (A.18) by considering a single training utterance and the indicator function

$$y_k^i = \frac{\sum_{t=1}^{T} I(\mathbf{z}_t \in i) z_{tk}}{\sum_{t=1}^{T} I(\mathbf{z}_t \in i)} \tag{A.26}$$

and turn the indicator function into its probabilistic version, the state occupancy $\gamma(t, i)$

$$y_k^i = \frac{\sum_{t=1}^{T} \gamma(t, i) z_{tk}}{\sum_{t=1}^{T} \gamma(t, i)} \tag{A.27}$$

HMM/SKL Model

Since the procedure for estimating the state distribution using the HMM/SKL is based on an iterative algorithm that only requires the geometric and arithmetic means of the training samples, we can use the means derived in (A.25) and (A.27) as input for this algorithm.

A.4 Maximum Likelihood Interpretation of HMM/KL

Let $Z(i)$ be the set of posterior features assigned to state i. Similarly, let us define $X(i)$ the set of acoustic features that yields $Z(i)$, i.e. for each $\mathbf{z} \in Z(i)$ there is an acoustic vector $\mathbf{x} \in X(i)$ such that $\mathbf{z} = P(\mathcal{C}|\mathbf{x})$. The set $\mathcal{C}$ denotes the set of posterior classes. In this case, we can reformulate (6.12) as

$$\begin{aligned}
\mathbf{y}_i &= \arg\min_{\mathbf{y}} \sum_{\mathbf{x}_n \in X(i)} KL(\mathbf{y}||P(\mathcal{C}|\mathbf{x}_n)) && \text{(A.28)}\\
&= \arg\min_{\mathbf{y}} \sum_{\mathbf{x}_n \in X(i)} \sum_{k=1}^{K} y_k \log \frac{y_k}{P(c_k|\mathbf{x}_n)} && \text{(A.29)}\\
&= \arg\min_{\mathbf{y}} \sum_{\mathbf{x}_n \in X(i)} \sum_{k=1}^{K} y_k \log \frac{y_k p(\mathbf{x}_n)}{p(\mathbf{x}_n|c_k)P(c_k)} && \text{(A.30)}\\
&= \arg\min_{\mathbf{y}} KL(\mathbf{y}||P(\mathcal{C})) - \int_{\mathcal{X}(i)} p(\mathbf{x}) \sum_{k=1}^{K} [y_k \log(p(\mathbf{x}|c_k)] \, d\mathbf{x} && \text{(A.31)}\\
&= \arg\max_{\mathbf{y}} \int_{\mathcal{X}(i)} p(\mathbf{x}) \sum_{k=1}^{K} [y_k \log p(\mathbf{x}|c_k)] \, d\mathbf{x} - KL(\mathbf{y}||P(\mathcal{C})) && \text{(A.32)}
\end{aligned}$$

where $\mathcal{X}(i)$ denotes the region in the acoustic space corresponding to the state i. In (A.31), the law of large numbers $\frac{1}{N}\sum_n f(\mathbf{x}_n) = E[f(\mathbf{x})]$ is applied. The constant factor N corresponding to the number of samples is removed because it does not affect to the minimization problem.

A.5 HMM/RKL Training Criterion

In this section, we develop the equality appearing in (6.15)

$$I(\mathcal{X},\mathcal{C}) - I(\hat{\mathcal{X}},\mathcal{C}) = \sum_{i=1}^{Q} \int_{\mathcal{X}(i)} p(\mathbf{x}) KL(\underbrace{P(\mathcal{C}|\mathbf{x})}_{\mathbf{z}} || \underbrace{P(\mathcal{C}|i)}_{\mathbf{y}_i}) \, d\mathbf{x} \qquad \text{(A.33)}$$

where $\mathcal{X}$ denotes a continuous set of the acoustic features and $\hat{\mathcal{X}}$ and $\mathcal{C}$ are discrete sets corresponding to states and phonemes respectively.

The left-handed term of the expression can be simplified by expanding the mutual information measure into a difference of entropies in the following way: $I(A,B) = H(A) - H(A|B)$. Then, the

left-handed term can be expressed as

$$
\begin{aligned}
I(\mathcal{X},\mathcal{C}) - I(\hat{\mathcal{X}},\mathcal{C}) &= -H(\mathcal{C}|\mathcal{X}) + H(\mathcal{C}|\hat{\mathcal{X}}) && \text{(A.34)}\\
&= \sum_{c_k} \int_{\mathcal{X}} p(c_k,\mathbf{x}) \log p(c_k|\mathbf{x}) d\mathbf{x} - \sum_{c_k}\sum_{i} p(c_k,i) \log p(c_k|i) && \text{(A.35)}\\
&= \sum_{c_k}\sum_{i} \int_{\mathcal{X}(i)} p(c_k,\mathbf{x}) \log p(c_k|\mathbf{x}) d\mathbf{x} - \sum_{c_k}\sum_{i} \int_{\mathcal{X}(i)} p(c_k,\mathbf{x}) \log p(c_k|i) d\mathbf{x} && \text{(A.36)}\\
&= \sum_{i} \int_{\mathcal{X}(i)} \sum_{c_k} p(c_k,\mathbf{x}) \log \frac{p(c_k|\mathbf{x})}{p(c_k|i)} && \text{(A.37)}\\
&= \sum_{i} \int_{\mathcal{X}(i)} p(\mathbf{x}) \sum_{c_k} p(c_k|\mathbf{x}) \log \frac{p(c_k|\mathbf{x})}{p(c_k|i)} && \text{(A.38)}\\
&= \sum_{i} \int_{\mathcal{X}(i)} p(\mathbf{x}) KL(p(\mathcal{C}|\mathbf{x})||p(\mathcal{C}|i) && \text{(A.39)}
\end{aligned}
$$

In step A.36, we consider that the space of acoustic features $\mathcal{X}$ can be split into a set of exclusive regions $\{\mathcal{X}(1),\cdots,\mathcal{X}(i),\cdots,\mathcal{X}(Q)\}$ where each region corresponds to one of the Q states of the model.

Bibliography

Abdou, S. and Scordilis, M. S. (2004). Beam Search Pruning in Speech Recognition Using a Posterior-based Confidence Measure. *Speech Communication*, **42**, 409–428.

Amari, S. (1967). A Theory of Adaptive Pattern Classifiers. *IEEE Transactions on Electronic Computers*, **3**, 299–307.

Amari, S. (2001). Information Geometry on Hierarchy of Probability Distributions. *IEEE Transactions on Information Theory*, **47**(5), 1701–1711.

Aradilla, G. and Bourlard, H. (2007). Posterior-Based Features and Distances in Template Matching for Speech Recognition. *4th Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms (MLMI)*.

Aradilla, G., Vepa, J., and Bourlard, H. (2005). Improving Speech Recognition Using a Data-Driven Approach. *Proceedings of Interspeech*, pages 3333–3336.

Aradilla, G., Vepa, J., and Bourlard, H. (2006a). Using Pitch as Prior Knowledge in Template-Based Speech Recognition. *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

Aradilla, G., Vepa, J., and Bourlard, H. (2006b). Using Posterior-Based Features in Template Matching for Speech Recognition. *Proceedings of International Conference on Spoken Language Processing (ICSLP)*.

Aradilla, G., Vepa, J., and Bourlard, H. (2007). An Acoustic Model Based on Kullback-Leibler Divergence for Posterior Features. *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, **4**, 657–660.

Aradilla, G., Bourlard, H., and Doss, M. M. (2008). Posterior Features Applied to Speech Recognition Tasks with Limited Training Data. *Submitted to Proceedings of International Conference on Spoken Language Processing (ICSLP)*.

Axelrod, S. and Maison, B. (2004). Combination of Hidden Markov Models with Dynamic Time Warping for Speech Recognition. *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, **I**, 173–176.

Bahl, L., Jelinek, F., and Mercer, R. (1983). A Maximum Likelihood Approach to Continuous Speech Recognition. *IEEE Transactions on Patten Analysis and Machine Intelligence (PAMI)*, **5**(2), 179–190.

Bahl, L. R., Brown, P. F., de Souza, P. V., and Mercer, R. L. (1986). Maximum Mutual Information Estimation of Hidden Markov Model Parameters. *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 49–52.

Bakis, R. (1976). Continuous Speech Recognition via Centisecond Acoustic States. *91st Meeting of the Acoustical Society of America*.

Bellman, R. (1966). Dynamic Programming. *Science*, **153**(3731), 34–37.

Bengio, Y. (1993). A Connectionist Approach to Speech Recognition. *International Journal of Pattern Recognition*, **7**(4), 647–667.

Berger, A. L., Pietra, S. A. D., and Pietra, V. J. D. (1996). A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, **22**(1), 39–71.

Berger, T. (1971). *Rate Distortion Theory: A Mathematical Basis for Data Compression*. Prentice-Hall.

Bisani, M. and Ney, H. (2004). Bootstrap Estimates for Confidence Intervals in ASR Performance Evaluation. *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, **1**, 409–412.

Black, A., Lenzo, K., and Pagel, V. (1998). Issues in Building General Letter to Sound Rules. *Proceedings of ESCA Workshop on Speech Synthesis*, pages 77–80.

Bourlard, H. and Morgan, N. (1993). *Connectionist Speech Recognition: A Hybrid Approach*, volume 247. Kluwer Academic Publishers, Boston.

Bourlard, H. and Wellekens, C. J. (1986). Connected Speech Recognition by Phonemic Semi-Markov Chains for State Occupancy Modeling. *Proceedings of EUSIPCO*, **1**, 511–514.

Bourlard, H. and Wellekens, C. J. (1990). Links Between Markov Models and Multilayer Perceptrons. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12**(12), 1167–1178.

Bourlard, H., Morgan, N., Wooters, C., and Renals, S. (1992). CDNN: A Context Dependent Neural Network for Continuous Speech Recognition. *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, **2**, 349–352.

Bourlard, H., Bengio, S., Doss, M. M., Zhu, Q., Mesot, B., and Morgan, N. (2004). Towards Using Hierarchical Posteriors for Flexible Automatic Speech Recognition Systems. *DARPA RT-04 Workshop*.

Breiman, L., Friedman, J., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees*. The Wadsworth Statistics/Probability Series.

Bridle, J. S. (1989). Probabilistic Scoring for Back-Propagation Networks, with Relationships to Statistical Pattern Recognition. *Proceedings of Neural Network for Computing*.

Bridle, J. S., Brown, M. D., and Chamberlain, R. M. (1983). Continuous Connected Word Recognition Using Whole Word Templates. *The Radio and Electronic Engineer*, **53**(4), 167–175.

Carpineto, C., Mori, R. D., Romano, G., and Bigi, B. (2001). An Information-Theoretic Approach to Automatic Query Expansion. *ACM Transactions on Information Systems*, **19**(1), 1–27.

Cerf, P. L., Weiye, M., and Compernolle, D. V. (1994). Multilayer Perceptrons as Labelers for Hidden Markov Models. *IEEE Transactions on Speech and Audio Processing*, **2**(1), 185–193.

Chen, B., Zhu, Q., and Morgan, N. (2004). Learning Long-term Temporal Features in LVCSR Using Neural Networks. *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, pages 925–928.

Cohen, M., Franco, H., Morgan, N., Rumelhart, D., and Abrash, V. (1993). Context-Dependent Multiple Distribution Phonetic Modelling with MLPs. *Advances in Neural Information Processing Systems*, pages 649–657.

Cole, R., Fanty, M., M., N., and Lander, T. (1995). New Telephone Speech Corpora at CSLU. *Proceedings of Eurospeech*, pages 821–824.

Cover, T. M. (1965). Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition. *IEEE Transactions on Electronic Computers*, **14**(3), 326–334.

Cover, T. M. and Thomas, J. A. (1991). *Information Theory*. John Wiley.

Davis, S. B. and Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Audio, Speech and Signal Processing*, **28**, 357–366.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data Via the EM algorithm. *J. Royal Statist. Soc. Ser. B*, **39**, 1–38.

Doss, M. M., Stephenson, T. A., and Bourlard, H. (2003). Using Pitch Frequency Information in Speech Recognition. *Proceedings of Eurospeech*, pages 2525–2528.

Duda, R. O., Hart, P. E., and Stork, D. G. (2001). *Pattern Classification*. Wiley-Interscience.

Dupont, S., Bourlard, H., Deroo, O., Fontaine, V., and Boite, J.-M. (1997). Hybrid HMM/ANN Systems for Training Independent Tasks: Experiments on Phonebook and Related Improvements. *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, **3**, 1767–1770.

Efron, B. and Tibshirani, M. J. (1993). *An Introduction to the Bootstrap*. Chapman & Hall.

Ellis, D. and Morgan, N. (1999). Size Matters: An Empirical Study of Neural Network Training for Large Vocabulary Continuous Speech Recognition. *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, **2**, 1013–1016.

Faria, A. and Morgan, N. (2008). Corrected Tandem Features for Acoustic Model Training. *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4737–4740.

Fosler-Lussier, E. and Morris, J. (2008). Crandem Systems: Conditional Random Field Acoustic Models for Hidden Markov Models. *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4049–4052.

Fritsch, J. and Finke, M. (1998). ACID/HNN: Clustering Hierarchies of Neural Networks for Context-dependent Connectionist Acoustic Modeling. *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, **1**, 505–508.

Fu, K. S. (1968). *Sequential Methods in Pattern Recognition and Machine Learning*. Academic Press.

Fukunaga, K. (1990). *Introduction to Statistical Pattern Recogntion*. Morgan Kaufmann, Academic Press.

Furui, S. (1986). Speaker-Independent Isolated Word Recognition Using Dynamic Features of Speech Spectrum. *IEEE Transactions on Acoustics, Speech and Signal Processing*, **34**, 52–59.

Ganapathiraju, A., Hamaker, J., and Picone, J. (2000). Hybrid SVM/HMM Architectures for Speech Recognition. *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, pages 504–507.

Gauvin, J.-L. and Lee, C.-H. (1992). MAP Estimation of Continuous Density HMM: Theory and Applications. *Proceedings of the Workshop on Speech and Natural Language*, pages 185–190.

Godfrey, J. J., Holliman, E. C., and McDaniel, J. (1992). Switchboard: Telephone Speech Corpus for Research and Developement. *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, **1**, 517–520.

Gokcay, E. and Principe, J. C. (2002). Information Theoretic Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**(2), 158–171.

Gold, B. and Morgan, N. (1999). *Speech and Audio Signal Processing: Processing and Perception of Speech and Music*. Wiley.

Goldinger, S. D. (1998). Echoes of Echoes? An Episodic Theory of Lexical Access. *Psychological Review*, **105**(2), 251–279.

Grezl, F., Karafiat, M., Kontar, S., and Cernocky, J. (2007). Probabilistic and Bottle-neck Features for LVCSR of Meetings. *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, **4**, 757–760.

Gunawardana, A., Mahajan, M., Acero, A., and Platt, J. C. (2005). Hidden Conditional Random Fields for Phone Classification. *Ninth European Conference on Speech Communication and Technology*.

Hennebert, J., Ris, C., Bourlard, H., Renals, S., and Morgan, N. (1997). Estimation of Global Posteriors and Forward-Backward Training of Hybrid HMM/ANN Systems. *Proceedings of Eurospeech*, pages 1951–1954.

Hermansky, H. (1990). Perceptual Linear Predictive (PLP) Analysis of Speech. *The Journal of the Acoustic Society of America*, **87**(4), 1738–1752.

Hermansky, H. and Fousek, P. (2005). Multi-Resolution RASTA Filtering for TANDEM-based ASR. *Proceedings of Interspeech*, pages 361–364.

Hermansky, H. and Sharma, S. (1999). Temporal Patterns (TRAPs) in ASR Noisy Speech. *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, **1**, 289–292.

Hermansky, H., Tibrewala, S., and Pavel, M. (1996). Towards ASR on Partially Corrupted Speech. *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, **1**, 462–465.

Hermansky, H., Ellis, D., and Sharma, S. (2000a). Tandem Connectionist Feature Extraction for Conventional HMM Systems. *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, **3**, 1635–1638.

Hermansky, H., Sharma, S., and Jain, P. (2000b). Data-derived Nonlinear Mapping for Feature Extraction in HMM. *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Hifny, Y. and Renals, S. (2005). A Hybrid MaxEnt/HMM based ASR System. *Proceedings of Interspeech*, pages 3017–3020.

Hochberg, M., Cook, G., Renals, S., Robinson, A., and Schechtman, R. (1995). Abbot hybrid connectionist hmm large-vocabulary recognition system. *Spoken Language Systems Technology Workshop*, pages 170–176.

Hunt, A. and Black, A. (1996). Unit Selection in a Concatenative Speech Synthesis System Using a Large Database. *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 373–376.

Illina, I. and Gong, Y. (1998). Elimination of Trajectory Folding Phenomenon: HMM, Trajectory Mixture HMM and Mixture Stochastic Trajectory Model. *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, **2**, 1395–1398.

Iwamida, H., Katagiri, S., and McDermott, E. (1991). Speaker-independent Large Vocabulary Word Recognition Using an LVQ/HMM hybrid algorithm. *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 553–556.

Jaynes, E. T. (1982). On the Rationale of Maximum-Entropy Methods. *Proceedings of IEEE*, **70**(9), 939–952.

Jelinek, F. (1976). Continuous Speech Recognition by Statistical Methods. *Proceedings of IEEE*, pages 532–556.

Jelinek, F. (2001). *Statistical Methods for Speech Recognition*. The MIT Press.

Joost, M. and Schiffmann, W. (1998). Speeding Up Backpropagation Algorithms by Using Cross-entropy Combined with Pattern Normalization. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems (IJUKFS)*, **6**(2), 117–126.

Juang, B. H. and Rabiner, L. R. (1990). The Segmental k-Means Algorithm for Estimating Parameters of Hidden Markov Models. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **38**, 1639–1641.

Katz, S. M. (1987). Estimation of Probabilities from Sparse Data for Language Model Component of a Speech Recognizer. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **35**(3), 400–401.

Kemperman, J. H. B. (1967). *On the Optimum Rate of Transmitting Information*. Springer-Verlag.

Klabbers, E. and Veldhuis, R. (2001). Reducing Audible Spectral Discontinuities. *IEEE Transactions on Speech and Audio Processing*, **9**(1), 39–51.

Klatt, D. H. (1977). Review of the ARPA Speech Understanding Project. *The Journal of the Acoustic Society of America*, **62**(6), 1345–1366.

Kohonen, T. (1988). *Self-organization and Associative Memory*. Springer.

Konig, Y. and Morgan, N. (1992). GDNN: a Gender-dependent Neural Network for Continuous Speech Recognition. *International Joint Conference on Neural Networks (IJCNN)*, **2**, 332–337.

Kullback, S. and Leibler, R. A. (1951). On Information and Sufficiency. *Annals of Mathematical Statistics*, **22**(1), 79–86.

Kuncheva, L. I. (2002). A Theoretical Study on Six Classifier Fusion Strategies. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**(2), 281–286.

Kuo, H.-K. J. and Gao, Y. (2006). Maximum entropy direct models for speech recognition. *IEEE Transactions on Audio Speech and Language Processing*, **14**(3), 873–881.

Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *International Conference on Machine Learning (ICML)*, pages 282–289.

Lee, K., Hon, H., and Reddy, R. (1990). An Overview of the SPHINX Speech Recognition System. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, pages 599–609.

Lefèvre, F. (2003). Non-Parametric probability estimation for HMM-Based Automatic Speech Recognition. *Computer Speech and Language*, **17**, 113–136.

Leung, H. C., Hetherington, L., and Zue, V. W. (1992). Speech Recognition Using Stochastic Segment Neural Networks. *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, **1**, 613–616.

Likhododev, A. and Gao, Y. (2002). Direct Models for Phoneme Recognition. *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 89–92.

Linde, Y., Buzo, A., and Gray, R. (1980). An Algorithm for Vector Quantizer Design. *IEEE Transactions on Communications*, **28**(1), 84–95.

Lippmann, R. P. and Gold, B. (1987). Neural Net Classifiers Useful for Speech Recognition. *IEEE Proceedings of International Conference on Neural Networks*, pages 417–425.

Ma, W. and Compernolle, D. V. (1990). TDNN Labeling for a HMM Recognizer. *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, **1**, 421–423.

Macherey, W. and Ney, H. (2003). A Comparative Study on Maximum Entropy and Discriminative Training for Acoustic MOdeling in Automatic Speech Recognition. *Proceedings of Eurospeech*, pages 493–496.

Makhoul, J. (1975). Linear Prediction: A Tutorial Review. *Proceedings of the IEEE*, **63**(4), 561–580.

Mangu, L., Brill, E., and Stolcke, A. (2000). Finding Consensus in Speech Recognition: Word Error Minimization and other Applications of Confusion Networks. *Computer, Speech and Language*, **14**, 373–400.

Mariethoz, J. and Bengio, S. (2004). A New Speech Recognition Baseline System for Numbers 95 Version 1.3 Based on Torch. Technical report, IDIAP Research Institute.

McCallum, A., Freitag, D., and Pereira, F. (2000). Maximum Entropy Markov Models for Information Extraction and Segmentation. *Proceedings of Machine Learning*, pages 591–598.

McDermott, E. and Katagiri, S. (1994). Prototype-based Minimum Classification Error/Generalized Probabilistic Descent Training for Various Speech Units. *Computer Speech and Language*, **186**(4), 351–368.

Merhav, N., Y., and Ephraim (1991). Hidden Markov modeling Using the Most Likely State Sequence. *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, **1**, 469–472.

Mika, S., Ratsch, G., Wetson, J., Scholkopf, B., and Mullers, K. R. (1999). Fisher Discriminant Analysis with Kernels. *IEEE Proceedings Neural Networks for Signal Processing*, pages 41–48.

Misra, H., Bourlard, H., and Tyagi, V. (2003). New Entropy Based Combination Rules in HMM/ANN Multi-stream ASR. *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Moon, T. K. (1996). The Expectation-Maximization Algorithm. *Proceedings of the IEEE*, **13**(9), 47–60.

Moore, B. C. J. (1997). *An Introduction to the Psychology of Hearing*. Academic Press.

Nocerino, N., Soong, F. K., Rabiner, L. R., and Klatt, D. H. (1985). Comparative Study of Several Distorsion Measures for Speech Recognition. *Proceedings of IEEE*, pages 25–28.

O'Brien, S. M. (1993). Knowledge-based Systems in Speech Recognition: a Survey. *Int. J. Man-Machine Studies*, **38**, 71–95.

O'Shaugnessy, D. (2003). Interacting with Computers via Voice: Automatic Speech Recognition and Synthesis. *Proceedings of IEEE*, **91**(9), 1272–1305.

Ostendorf, M., Digalakis, V. V., and Kimball, O. A. (1996). From HMM's to Segment Models: A Unified View of Stochastic Modeling for Speech Recognition. *IEEE Transactions on Speech and Audio Processing*, **4**(5), 360–378.

Paul, D. B. and Baker, J. M. (1992). The Design for the Wall Street Journal-based CSR Corpus. *DARPA Speech and Language Workshop*.

Pinto, J., Yegnanarayana, B., Hermansky, H., and Doss, M. M. (2008). Exploiting Contextual Information for Improved Phoneme Recognition. *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Pitrelli, J., Fong, C., Wong, S., Spitz, J., and Leung, H. (1995). Phonebook: A Phonetically-rich Isolated-word Telephone-speech Database. *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 101–104.

Platt, J. (2000). *Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods*. MIT Press.

Povey, D., Kingbury, B., Mangu, L., Saon, G., Soltau, H., and Zweig, G. (2005). fMPE: Discriminatively Trained Features for Speech Recognition. *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, **1**, 961–964.

Price, P. J., Fisher, W., and Bernstein, J. (1988). A database for continuous speech recognition in a 1000 word domain. *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 651–654.

Rabiner, L. and Juang, B. H. (1993). *Fundamentals of Speech Recognition*. Prentice Hall.

Rabiner, L. R. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, pages 257–286.

Rabiner, L. R. and Schafer, R. W. (1978). *Digital Processing of Speech Signals*. Prentice Hall.

Renals, S., Morgan, N., Bourlard, H., Cohen, M., and Franco, H. (1994). Connectionist Probability Estimators in HMM Speech Recognition. *IEEE Transactions on Speech and Audio Processing*, **2**(1), 161–174.

Reyes-Gomez, M. J. and Ellis, D. P. W. (2002). Error Visualization for Tandem Acoustic Modeling on the Aurora Task. *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, **4**, 4176–4179.

Richard, M. D. and Lippmann, R. P. (1991). Neural Network Classifiers Estimate Bayesian a posteriori Probabilities. *Neural Computation*, **3**, 461–483.

Rigoll, G. (1994). Maximum Mutual Information Neural Networks for Hybrid Connectionist HMM Speech Recognition Systems. *IEEE Transactions on Speech and Audio Processing*, **2**(1), 175–184.

Robinson, T. (1994). An Application of Recurrent Nets to Phone Probability Estimation. *IEEE Transactions of Neural Networks*, **5**, 298–305.

Rottland, J. and Rigoll, G. (2000). Tied Posteriors: An Approach for Effective Introduction of Context Dependency in Hybrid NN/MLP LVCSR. *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, **3**, 1241–1244.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning Internal Representations by Error Propagation. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*, pages 318–362.

Sakoe, H. and Chiba, S. (1978). Dynamic Programming Algorithm Optimization for Spoken Word Recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, **26**(1), 43–49.

Salomon, J., King, S., and Osborne, M. (2002). Framewise Phone Classification Using Support Vector Machines. *Proceedings of International Conference on Spoken Language Processing (ICSLP)*.

Sankoff, D. and Kruskal, J. (1999). *Time Warps, String Edits and Macromolecules: The theory and practise of sequence comparison*. CSLI Publications, Leland Stanford Junior University.

Santini, S. and Braun, H. (1995). Recurrent Neural Networks Can Be Trained to Be Maximum a Posteriori Probability Classifiers. *Neural Networks*, **8**(1), 25–29.

Shannon, C. E. (1948). A Mathematical Theory of Communication. *The Bell System Technical Journal*, **27**, 623–656.

Shire, M. L. (2001). Relating Frame Accuracy with Word Error in Hybrid ANN-HMM ASR. *Proceedings of Eurospeech*, pages 1797–1800.

Sim, K. C. and Gales, M. J. F. (2007). Discriminative Semi-parametric Trajectory Model for Speech Recognition. *Computer Speech and Language*, **21**(4), 669–687.

Stevens, S. S. (1957). On the Psychophysical Law. *Psychological Review*, **64**, 153–181.

Strik, H. (2003). Speech is like a box of chocolates... *Proceedings of 15th ICPhS*, pages 227–230.

Taylor, P., Black, A. W., and Caley, R. (1998). The Architecture of the Festival Speech Synthesis System. *ESCA/OSCODA Workshop on Speech Synthesis*, pages 147–152.

Tishby, N., Pereira, F. C., and Bialek, W. (1999). The Information Bottleneck Method. *Proceedings of the 37th Annual Allerton Conference on Communication, Control and Computing*, pages 368–377.

Tsuboka, E. and Nakahashi, J. (1994). On the Fuzzy Vector Quantization Based Hidden Markov Model. *IEEE Transactions on Speech and Audio Processing*, **1**, 637–640.

Valente, F. and Hermansky, H. (2007). Combination of Acoustic Classifiers Based on Dempster-Shafer Theory of Evidence. *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, **4**, 1129–1132.

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer.

Veldhuis, R. (2002). The Centroid of the Symmetrical Kullback-Leibler Distance. *IEEE Signal Processing Letters*, **9**(3), 96–99.

Vepa, J., King, S., and Taylor, P. (2002). Objective Distance Measures for Spectral Discontinuities in Concatenative Speech synthesis. *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, pages 2605–2608.

Vinga, S. and Almeida, J. (2003). Alignment-free Sequence Comparison - A review. *Bioinformatics*, **19**(4), 513–523.

Viterbi, A. J. (1967). Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm. *IEEE Transactions on Information Theory*, **13**, 260–269.

Wachter, M. D., Demuynck, K., Compernolle, D. V., and Wambacq, P. (2003). Data Driven Example Based Continuous Speech Recognition. *Proceedings of Eurospeech*, pages 1133–1136.

Wachter, M. D., Matton, M., Demuynck, K., Wambacq, P., Cools, R., and Compernolle, D. V. (2007). Template-Based Continuous Speech Recognition. *IEEE Transactions on Audio, Speech and Signal Processing*, **15**(4), 1377–1390.

White, H. (1990). Multilayer Feedforward Networks Can Learn Arbitrary Mappings. *Neural Networks*, **3**(5), 535–549.

William, G. and Renals, S. (1999). Confidence Measures from Local Posterior Estimate. *Computer, Speech and Language*, **13**, 395–411.

Wouters, J. and Macon, M. W. (1998). A Perceptual Evaluation of Distance Measures for Concatenative Speech Synthesis. *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, pages 2747–2750.

Yang, H., van Vuuren, S., and Hermansky, H. (1999). Relevancy of Time-frequency Features for Phonetic Classification of Phonemes. *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, **1**, 225–229.

Young, S. J., Odell, J. J., and Woodland, P. C. (1994). Tree-base State Tying for High Accuracy Acoustic Modeling. *Proceedings of the Workshop on Human Language Technology*.

Curriculum Vitae

Guillermo Aradilla Zapata

Permanent address:	Calle de la Marina, 104 6 2	Phone:	+41 764 23 72 28
	08018 - Barcelona, Spain	email:	aradilla@idiap.ch
		Citizenship:	Spanish

Education

April 2004–	Docteur ès Sciences (anticipated August 2008). The Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland. Thesis title: *Acoustic Models for Posterior Features in Speech Recognition*.
March 2002–April 2004	European Master in Language and Speech Universitat Politècnica de Catalunya (UPC), Barcelona, Switzerland.
Sep. 1998–April 2004	Bachelor & Master of Electrical Engineering Universitat Politècnica de Catalunya (UPC), Barcelona, Switzerland.

Professional Experience

April 2007–July 2007	Visit at Deutsche Telekom Laboratories (Berlin, Germany) Detection of telephone numbers in voice messages
Sep. 2003–June 2008	Research Assistant at IDIAP Research Institute (Martigny, Switzerland) Machine learning approaches for automatic speech recognition
Oct. 2006–March 2007	Supervision of Master project student (Martigny, Switzerland) Real time speech recognition system
May 2003–Sep. 2003	Internship at Indra (Barcelona, Spain) Evaluation of protocols for satellite communications
Feb. 2003–May 2003	Internship at Jazztel (Barcelona, Spain) Management of landlines
Oct. 2001–Feb. 2003	Internship at Retevision (Barcelona, Spain) Control of radio base stations

Publications

Book Chapters and Publications on Journals

- Guillermo Aradilla and Hervé Bourlard, (2007). Posterior-based Features and Distances in Template Matching for Speech Recognition, Machine Learning for Multimodal Interaction (MLMI), *Lecture Notes in Computer Science*, Volume No. 4892, Springler-Verlag.

- Guillermo Aradilla, Hervé Bourlard and Mathew Magimai Doss (2007). Kullback-Leibler Divergence Based Acoustic Models for Posterior Features in Speech Recognition, *Submitted to IEEE Transactions on Audio, Speeech and Language Processing*.

Publications in International Conferences

- Guillermo Aradilla, Hervé Bourlard and Mathew Magimai Doss (2007). Using KL-based Acoustic Models in a Large Vocabulary Recognition Task. *Submitted to International Conference on Spoken Language Processing (ICSLP)*.

- Guillermo Aradilla, Hervé Bourlard and Mathew Magimai Doss (2007). Posterior Features Applied to Speech Recognition Tasks with Limited Training Data. *Submitted to International Conference on Spoken Language Processing (ICSLP)*.

- Guillermo Aradilla and Jitendra Ajmera (2007). Detection and Recognition of Number Sequences in Spoken Utterances. *2nd Workshop on Speech in Mobile and Pervasive Environments (SiMPE)*.

- Guillermo Aradilla, Jitendra Vepa and Hervé Bourlard (2007). An Acoustic Model Based on Kullback-Leibler Divergence for Posterior Features. *Proceedings of IEEE International Conference on Acoustic Speech Signal Processing (ICASSP)*

- Guillermo Aradilla, Jitendra Vepa and Hervé Bourlard (2006). Using Posterior-Based Features in Template Matching for Speech Recognition. *Proceedings of International Conference on Spoken Language Processing (ICSLP)*.

- Guillermo Aradilla, Jitendra Vepa and Hervé Bourlard (2006). Using Pitch as Prior Knowledge in Template-Based Speech Recognition. *Proceedings of IEEE International Conference on Acoustic Speech Signal Processing (ICASSP)*

- Guillermo Aradilla, Jitendra Vepa and Hervé Bourlard (2005). Improving Speech Recognition Using a Data-Driven Approach. *Proceedings of Interspeech*

- Guillermo Aradilla, John Dines and Sunil Sivadas (2004). Using RASTA in Task Independent TANDEM Feature Extraction. *Proceedings of International Conference on Spoken Language Processing (ICSLP)*.